



Behavioral approach for validation and system testing of embedded systems : Application in medical embedded devices

Charbel El Gemayel

► To cite this version:

Charbel El Gemayel. Behavioral approach for validation and system testing of embedded systems : Application in medical embedded devices. Quantitative Methods [q-bio.QM]. INSA de Lyon; Université Libanaise, 2014. English. NNT : 2014ISAL0135 . tel-01339369

HAL Id: tel-01339369

<https://theses.hal.science/tel-01339369>

Submitted on 29 Jun 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THESE EN COTUTELLE

Pour obtenir le grade de Docteur délivré par

L'Institut National des

Sciences Appliquées de Lyon

Et

L'Université Libanaise

Ecole Doctorale des Sciences et Technologie

Spécialité : Electronique - Informatique

Présentée et soutenue publiquement par

El Gemayel Charbel

17/12/2014

**Approche comportementale pour la validation et le test système des
systèmes embarqués; application aux dispositifs médicaux embarqués**

Directeurs de thèse : **Abouchi Nacer – Zaouk Doumit**

Co-encadrement de la thèse : **Jumel Fabrice – Constantin Joseph**

Membre du Jury

M. Toni Sayah, Professeur, Université Saint Joseph

M. Richard Grisel, Professeur, Université de Rouen

M. Cyril Condemine, Docteur, CEA-LETI, Grenoble

M. Youssef Zaatari, Professeur, Université Libanaise

M. Fabrice Jumel, Maître de conférences, CPE Lyon

M. Joseph Constantin, Maître de conférences, Université Libanaise

M. Doumit Zaouk, Professeur, Université Libanaise

M. Nacer Abouchi, Professeur, CPE Lyon

Rapporteur

Rapporteur

Examineur

Examineur

Co-directeur

Co-directeur

Directeur

Directeur

**Thèse préparée au laboratoire CPE-INSA Lyon et au laboratoire LPA-Faculté des
Sciences II de l'Université Libanaise.**

COTUTELLE THESIS

For the degree of Doctor delivered by

NATIONAL INSTITUTE FOR APPLIED

SCIENCES OF LYON

AND

LEBANESE UNIVERSITY

DOCTORAL SCHOOL OF SCIENCE AND TECHNOLOGY

Speciality: Electronics - Informatics

Presented and publicly supported by

El Gemayel Charbel

17/12/2014

**Behavioral approach for validation and system testing of
embedded systems; application in medical embedded devices**

Thesis director : **Abouchi Nacer – Zaouk Doumit**

Thesis Co-director: **Jumel Fabrice – Constantin Joseph**

Membre du Jury

M. Toni Sayah, Professor, Saint Joseph University

M. Richard Grisel, Professor, Rouen University

M. Cyril Condemine, Doctor, CEA-LETI, Grenoble

M. Youssef Zaatara, Professor, Lebanese University

M. Fabrice Jumel, Associate Professor, CPE Lyon

M. Joseph Constantin, Associate Professor, Lebanese University

M. Doumit Zaouk, Professor, Lebanese University

M. Nacer Abouchi, Professor, CPE Lyon

Report examiner

Report examiner

Examiner

Examiner

Co-director

Co-director

Director

Director

The thesis is prepared in the laboratory of CPE-INSAL Lyon and LPA-Faculty of Sciences II at the Lebanese University.

Acknowledgments

There are many people who helped me during my studies and this work would not end without their contributions.

First of all, I wish to express my deep gratitude to my supervisor's Prof Nacer Abouchi, Prof Doumit Zaouk, Dr. Fabrice Jumel and Dr. Joseph Constantin for their advice and help within this research. I thank them for their trust they gave me, by allowing me to do a thesis under their direction. Their advice and assistance have been most helpful to the success of this work.

I have the honor to do a Ph.D. under joint supervision between the National Institute of Applied Sciences (INSA) in Lyon and the Lebanese University in collaboration with the INL Laboratory (CPE Lyon) and LPA laboratory Fanar.

I would like to thank the report examiners (i.e. rapporteurs): Prof Richard Grisele and Prof Toni Sayah for having accepted and efficiently reviewed this work. Also, I would like to thank the examiners Dr. Cyril Condemine, Dr. Youssef Zaatar. Furthermore, I thank all members of the jury for their presence, and I am very grateful to them for their helpful comments and feedback.

Finally I would like to thank Pr. Antonio Khoury, without him I am not now writing this acknowledgement, and the director of the Lebanese university- faculty of sciences II Fanar- Dr. Georges Rahbani for his support. I would like to thank my parents and family for all of their support and encouragement during these years of research.

Abstract

A Biomedical research seeks good reasoning for solving medical problems, based on intensive work and great debate. It often deals with beliefs or theories that can be proven, disproven or often refined after observations or experiments. The problem is how to make tests without risks for patients, including variability and uncertainty on a number of parameters (patients, evolution of disease, treatments ...). Nowadays, medical treatment uses more and more embedded devices such as sensors, actuators, and controllers. Treatment depends on the availability and well-functioning of complex electronic systems, comprising thousands of lines of codes. A mathematical representation of patient or device is presented by a number of variables which are defined to represent the inputs, the outputs and a set of equations describing the interaction of these variables.

The objective of this research is to develop tools and methodologies for the development of embedded systems for medical fields. The goal is to be able to model and jointly simulate the medical device as well the human body, at least the part of the body involved in the medical device, to analyze the performance and quality of service (QoS) of the interaction of the device with the human body. To achieve this goal our study focused on several points described below.

After starting by defining a prototype of a new global and flexible architecture of mathematical model of human body, which is able to contain required data, we begin by proposing a new global methodology for modeling and simulation human body and medical systems, in order to better understand the best way to model and simulate these systems and for detecting performance and the quality of services of all system components. We use two techniques that help to evaluate the calculated QoS value. The first one calculates an index of severity which indicates the severity of the case studied. The second one using a normalization function that represents the simulation as a point in order to construct a new error grid and use it to evaluate the accuracy of value measured by patients.

Using Keil development tools designed for ARM processors, we have declared a new framework in the objective to create a new tester model for the glucose-insulin system, and to define the basic rules for the tester which has the ability to satisfy well-established medical decision criteria. The advantage of using Keil in our work is to define a model that works with embedded C language and can be implemented in a microcontroller.

The framework begins by simulating a mathematical model of the human body, and this model was developed to operate in the closed loop of the glucose insulin. Then, the model of artificial pancreas has been implemented to control the mathematical model of human body. Finally a new tester model was created in order to analyze the performance of all the components of the glucose-insulin system.

As we know, the in-silico experimentation presents many advantages by providing higher work productivity, minimum cost, and more accurate simulations through more sophisticated models. Then, in the simulation part we use a mathematical model “Hovorka” which is tested on an in silico trial of type I diabetic subjects for 4 days, receiving breakfast, lunch, and dinner each day. Performance of the algorithm was analyzed with the consensus error grid using data sets generated by virtual patients and parameters changes. Then we analyze each simulation in purpose to have a graphical representation of the risk assessment for the patient due to choices on sensors, actuators.

By generating virtual patients, we had generalized the passage of the clinical trial in order to have several scenarios for analyzing performance. The objective is to have multiple scenarios in order to use them in the implementation part. Taking into consideration the approximation range of each parameter, we have created a simple tool to generate many patients. A simple modification in each parameters help to have different patient states. The severity of a patient can be modified by changing parameters’ values. Indeed, solving mathematical equations inside each parameter helps to have “clinically accurate” results.

The simulation of mathematical models provides useful tools for diagnosis and analysis of the interactions between efficacy, therapies, side-effects, and outcomes. This will help us to better understand the human organism control, analyze experimental data, identify and quantify relevant biophysical parameters, and to design clinical trials. We have used the suitability of partially observable Markov decision processes to formalize the planning of clinical management. We have applied a virtual population of subjects with type 1 diabetes comprises a simulation model of the glucose regulation accompanied by N parameter sets representing N virtual subjects.

Keywords: modeling, simulation, medical systems, microcontroller, quality of services, architecture, mathematical model, diagnosis, control, in-silico, virtual patients, clinical case.

Résumé

Les progrès des technologies de l'information et de la communication, des MEMS, des capteurs, actionneurs, etc. ont permis l'émergence de différents dispositifs biomédicaux. Ces nouveaux dispositifs, souvent embarqués, contribuent considérablement à l'amélioration du diagnostic et du traitement de certaines maladies, comme le diabète par exemple. Des dispositifs embarqués encore plus complexes sont en cours d'élaboration, leur mise en œuvre nécessite des années de recherche et beaucoup d'expérimentation. Le cœur artificiel, encore en phase de réalisation, est un exemple concret de ces systèmes complexes. La question de la fiabilité, du test de fonctionnement et de sûreté de ces dispositifs reste problématique et difficile à résoudre. Plusieurs paramètres (patient, évolution de la maladie, alimentation, activité, traitement, etc.) sont en effet à prendre en compte et la conséquence d'une erreur de fonctionnement peut être catastrophique pour le patient. L'objectif de cette thèse est de développer des outils et des approches méthodologiques permettant la validation et le test au niveau système de ce type de dispositifs. Il s'agit précisément d'étudier la possibilité de modéliser et simuler d'une manière conjointe un dispositif médical ainsi que son interaction avec le corps humain, du moins la partie du corps humain concernée par le dispositif médical, afin de mesurer les performances et la qualité de services (QoS) du dispositif considéré.

Pour atteindre cet objectif notre étude a porté sur plusieurs points. Nous avons d'abord mis en évidence une architecture simplifiée d'un modèle de corps humain permettant de représenter et de mieux comprendre les différents mécanismes du corps humain. Nous avons ensuite exploré un ensemble de métriques et une approche méthodologique générique permettant de qualifier la qualité de service d'un dispositif médical donné en interaction avec le corps humain.

Afin de valider notre approche, nous l'avons appliquée à un dispositif destiné à la régulation du taux de sucre pour des patients atteints du diabète. La partie du corps humain concernée par cette pathologie à savoir le pancréas a été simulé par un modèle simplifié que nous avons implémenté sur un microcontrôleur. Le dispositif de régulation de l'insuline quant à lui a été simulé par un modèle informatique écrit en C. Afin de rendre les mesures de performances observées indépendantes d'un patient donné, nous avons étudiés différentes stratégies de tests sur différentes catégories de patients. Nous avons pour cette partie mis en œuvre un générateur de modèles capable de reproduire différents états physiologiques de patients diabétiques. L'analyse et l'exploitation des résultats observés peut aider les médecins à considérablement limités les essais cliniques sur des vrai patients et les focaliser uniquement sur les cas les plus pertinent.

Mots-clés: méthodologie, modélisation, simulation, dispositifs médicaux, qualité de services, architecture, modèle mathématique, diagnostic, analyse, contrôle, microcontrôleur, essais cliniques.

Introduction

Les progrès des technologies de l'information et de la communication, des MEMS, des capteurs [21, 22, 23], des actionneurs [24, 25], des microcontrôleurs [26], etc. ont permis l'émergence de différents dispositifs biomédicaux. Ces nouveaux dispositifs contribuent considérablement à l'amélioration du diagnostic et du traitement de certaines maladies, comme le diabète par exemple. Des dispositifs embarqués encore plus complexes sont en cours d'élaboration, leur mise en œuvre nécessite des années de recherche et beaucoup d'expérimentation [15, 16, 17]. Le cœur artificiel, encore en phase de réalisation, est un exemple concret de ces systèmes complexes. La question du test de fonctionnement et de sûreté de ces dispositifs reste problématique et difficile à résoudre. Plusieurs paramètres (patient, évolution de la maladie, traitement, etc.) sont en effet à prendre en compte et la conséquence sur le patient peut être catastrophique.

L'objectif des travaux de recherche de cette thèse est de développer des outils et des méthodologies permettant la validation et le test au niveau système de dispositifs embarqués pour le domaine médical. Le but étant de pouvoir modéliser et simuler d'une manière conjointe le dispositif médical ainsi que le corps humain, du moins la partie du corps humain concernée par le dispositif médical, afin d'analyser les performances et la qualité de services (QoS) du dispositif considéré en interaction avec le corps humain. Pour atteindre cet objectif notre étude a porté sur plusieurs points. Nous avons d'abord étudié une architecture simplifiée d'un modèle de corps humain permettant de représenter et de mieux comprendre différents mécanismes du corps humain. Nous avons ensuite proposé un ensemble de métriques et une approche méthodologique générique permettant de qualifier la qualité de service d'un dispositif médical donné en interaction avec le corps humain. Deux techniques d'aide à l'évaluation de qualité de services (QoS) sont mises en évidence.

La première technique calcule un indice de sévérité qui quantifie la gravité du cas étudié. La seconde, basée sur une fonction de normalisation relative à une grille d'erreur construite par simulation de différents scénarios, permet d'évaluer la précision de la valeur mesurée.

Afin de valider notre approche, nous l'avons appliquée sur le cas du diabète. La partie du corps humain concernée par cette pathologie à savoir le pancréas a été simulé par un modèle mathématique simplifié que nous avons implémenté sur un microcontrôleur. Le dispositif du contrôle de régulation et d'injection de l'insuline quant à lui a été simulé par un modèle informatique écrit en C. Nous avons pour cela développé, en utilisant les outils de développement Keil conçus pour les processeurs ARM, un modèle mathématique permettant le contrôle et le fonctionnement en boucle fermée du système glucose-insuline. Nous avons aussi définis les règles de base pour le dispositif à évaluer. Enfin, un nouveau modèle de contrôle a été créé afin d'analyser les performances de tous les composants du système glucose-insuline. Pour disposer de mesures fiables, nous avons étudié le cas de plusieurs patients auxquelles nous avons appliqués différentes stratégies de tests. Nous avons pour cette partie mis en œuvre un générateur de modèles de patients capable de reproduire différents états physiologiques d'un patient atteint du diabète de type 1 et ainsi étudier les performances du dispositif face à différents scénarios. Nous avons donc créé un outil simple pour générer de nombreux patients en prenant en compte les spécificités de chaque paramètre. Une simple modification dans chacun des paramètres aide à modéliser l'état d'un patient différent. L'état de gravité d'un patient peut ainsi être modifié en changeant les valeurs des paramètres du modèle mathématique, ceci permet d'avoir des résultats cliniquement précis.

Nous avons utilisé pour la simulation globale l'approche expérimentale in-silico qui présente de nombreux avantages et qui permet des simulations précises grâce à des modèles sophistiqués. Pour la partie simulation globale, nous avons utilisé le modèle mathématique "Hovorka" qui a été testé sur un ensemble de sujets diabétiques recevant un petit déjeuner, un

déjeuner et un dîner chaque jour pendant 4 jours. Les performances de ce modèle ont été analysées par comparaison à la grille d'erreur générées par simulation de différents patients dans différents scénarios. L'analyse de chaque simulation a été effectuée en vue d'avoir une représentation graphique de l'évaluation des risques pour le patient en raison des choix des capteurs et des actionneurs. L'analyse et l'exploitation des résultats observés aidera à mieux comprendre le contrôle de l'organisme humain pour identifier et quantifier les paramètres biophysiques pertinents et à considérablement limités les essais cliniques et les focaliser uniquement sur les cas les plus pertinents.

Le résumé est structuré comme suite : après l'introduction, nous présentons dans le premier chapitre notre étude sur la modélisation du corps humain. L'architecture du modèle retenu et la représentation mathématique du corps humain. Le deuxième chapitre sera consacré à la simulation des systèmes biomédicaux et l'évaluation de leur qualité de service. La structure du simulateur et de l'analyseur seront présentés ainsi que les schémas des indicateurs mis en œuvre pour l'évaluation de la Qualité de Service. Le chapitre 3 présente l'étude de cas considérée et les simulations mises en place. On présente dans ce chapitre, les modèles réalisés, les algorithmes de contrôles, la génération des modèles de patients virtuels et enfin les performances de contrôle en boucle fermée obtenus en exploitant l'approche expérimentale in-silico. Le dernier chapitre est consacré à la conclusion et aux perspectives. On reviendra sur les objectifs de la thèse et les réponses apportées.

Chapitre 1 : modélisation du corps humain

1.1. Introduction

La modélisation et la simulation du corps humain et de l'interaction de ses variables physiologiques avec des dispositifs médicaux fait partie des sujets les plus étudiés dans le domaine de la recherche. Dans le cadre de cette thèse, un dispositif médicale représente tout instrument, appareil, outil, machine, logiciel, matériel, etc., destiné être utilisé sur le corps humain, seul ou en combinaison avec d'autres dispositifs, pour une opération de diagnostic, de prévention, de contrôle, de traitement d'une maladie, etc. Dans ce chapitre, nous présentons une méthodologie pour la modélisation et la simulation globale de dispositifs médicaux et de leurs interactions avec le corps humain, afin d'analyser leurs performances et d'évaluer leurs qualités des services.

1.2. Rappel sur le fonctionnement du corps humain

Se déplacer, réagir à son environnement, ingérer et digérer de la nourriture, avoir une activité métabolique, éliminer les déchets, se reproduire, etc. fait appel à différentes fonctions vitales interdépendantes et fait de l'humain un système d'une extrême complexité [34, 35]. La figure 1 montre à titre d'illustration l'interdépendance de différentes parties du système corps humain, où le système tégumentaire protège l'ensemble du corps contre l'environnement. Une membrane forme une enveloppe et permet le passage des substances utiles, tout en empêchant le passage de substances nocives ou inutiles. Le système digestif et le système respiratoire interagissent avec l'environnement et fournissent respectivement les éléments nutritifs et l'oxygène au sang chargé de les distribuer à l'ensemble des cellules du corps. Les déchets métaboliques sont éliminés de l'organisme par l'intermédiaire du système urinaire et du système respiratoire. Aucun de ces systèmes ne travaille d'une manière totalement indépendante, ils travaillent tous ensemble, pour le bien-être de l'organisme entier (ex.. digestif, cardio-vasculaire, musculaire et respiratoire).

Pour garantir l'équilibre du corps humain, beaucoup de variables physiologiques (que nous notons par la suite PV) doivent être contrôlées et régulées. C'est par exemple le cas de la pression artérielle, de l'acidité du sang, du taux sanguin de sucre, de la fréquence cardiaque, de la température corporelle, du rythme respiratoire, etc.

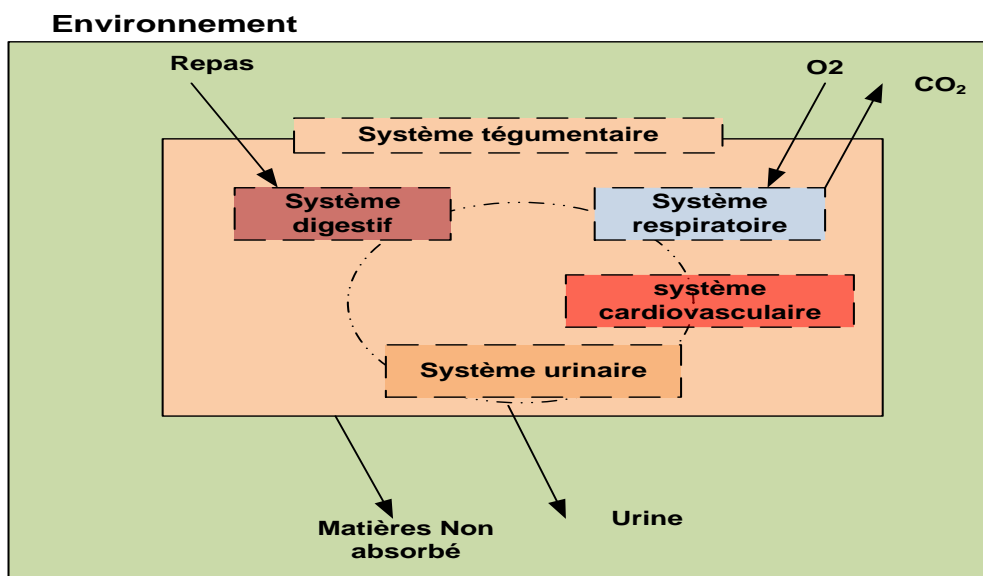


Fig. 1: Interdépendance des systèmes de l'organisme

Tout événement qui affecte le corps humain, peut provoquer un changement de variables physiologiques. En réaction le système « corps humain », s'il ne présente pas d'anomalies, corrige ces changements et maintient les variables physiologiques dans des limites acceptables propres à chaque individu. Prenons le cas de la pression artérielle qui est l'une des variables physiologiques contrôlées par le corps humain. Lorsqu'un événement externe ou interne provoque une augmentation de cette variable, agissant comme des capteurs les cellules nerveuses sensibles à la pression dans certaines artères envoient sous formes d'impulsions nerveuses des données au cerveau qui joue le rôle du contrôleur. Le cerveau interprète les messages et sort moins d'influx nerveux en direction des artérioles. Agissant comme des actionneurs, les artérioles se dilatent pour corriger la pression artérielle. La figure 2 montre comment ce mécanisme de régulation peut être assimilé et modélisé par un système de régulation automatique en boucle fermée.

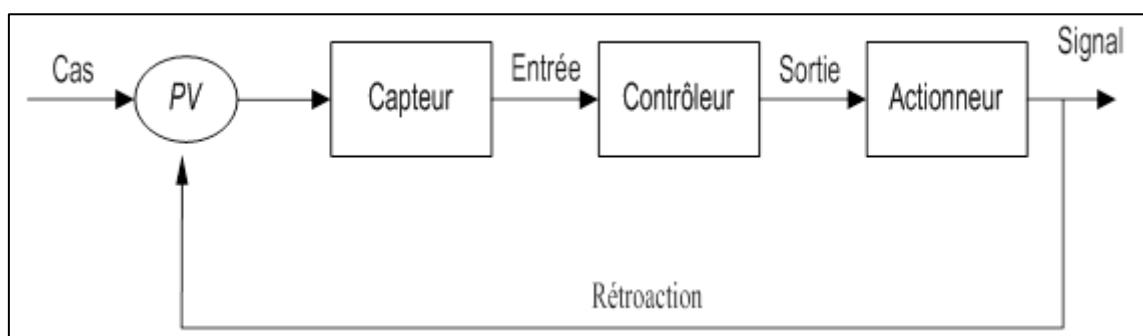


Fig. 2: Modèle de régulation d'une variable physiologique

Le contrôle et régulation d'une variable physiologique repose exactement sur ce modèle. Un capteur est chargé par la transmission des variations de la variable physiologique à surveiller à un contrôleur. Ce dernier connaissant la valeur de référence qui doit être maintenue, analyse les données reçût et détermine la réponse appropriée. A réception de cette réponse, un actionneur produit la contre-réaction positive ou négative nécessaire à la correction des variations du paramètre physiologique et sa stabilisation autour de la valeur considérée comme normale. La loi de commande est dans certains cas très complexe. Ces quelques rappels montrent combien la modélisation de tout ou partie du corps humain est complexe. Il s'agit comme mentionnée précédemment un des systèmes complexes. Cependant et en fonction des objectifs des études souhaitées, plusieurs niveaux de complexité peuvent être considérés et la modélisation peut être en partie simplifiée.

1.3. Représentation mathématique du corps humain

L'utilisation d'équations mathématiques pour la modélisation des systèmes biologiques, particulièrement humains, est exploitée depuis de longue date. La complexité du corps humain à même rendu cette approche incontournable pour la compréhension de certaines fonctionnalités du corps humain [36, 37]. Les modèles de distribution concernant l'espèce humaine ont en été et sont aujourd'hui utilisés dans de nombreux domaines [38]. Le modèle mathématique d'un patient est souvent basé sur un ensemble d'équations mathématiques et de divers algorithmes de contrôle-commande. Les paramètres et les constantes manipulés par ces équations et algorithmes reproduisent les entrées et sorties permettant de simuler le fonctionnement du corps humain.

Si Mp définit une fonction représentant le modèle mathématique d'un patient et si :

- S représente un ensemble de systèmes du corps humain (appareil digestif, appareil respiratoire, etc.).
- Pf représente un ensemble d'informations caractéristiques du patient (poids, stress, etc.).
- Dm représente ensemble des repas (le petit-déjeuner, le déjeuner, le goûter, le dîner, etc.).
- Or représente un ensemble d'organes du corps (foie, cœur, pancréas, etc.)
- Ir un ensemble de relations internes entre les organes du corps humain (Voir l'exemple ci-dessous)

- **Er** un ensemble des relations externes entre les organes du corps humain (Voir l'exemple ci-dessous)
- **C** et **PV** un ensemble de constantes et variables physiologiques liées aux systèmes et correspondant à l'état d'un patient, **C** et **PV** ont des valeurs numériques qui peuvent être utilisées par les équations du modèle. Notons **SC** un jeu de constantes sous-ensemble de **C** et **SPV** un jeu de variables physiologiques sous-ensemble de **PV**, spécifiques à un patient.
- les variables **Par** et **SPar** paramètres d'entrée pour le système, **Spar** sous-ensemble de **Par**

Si **W** représente le poids du patient, **St** un paramètre représentatif du stress du patient, **Sp** un paramètre indiquant si la personne fait du sport, etc. On peut définir l'ensemble d'informations personnelles sur le patient par :

$$\mathbf{Pf} = \{\mathbf{W}, \mathbf{St}, \mathbf{Sp} \dots\} \quad (1)$$

Si **B** représente l'ensemble des valeurs alimentaires du petit-déjeuner, comme les protéines de gras etc., **L** est l'ensemble de la valeur alimentaire du déjeuner, **D** l'ensemble des valeurs alimentaires pour le dîner et **Af** l'ensemble des portions supplémentaires tels que les collations. On définit l'ensemble des repas quotidiens par :

$$\mathbf{Dm} = \{\mathbf{B}, \mathbf{L}, \mathbf{D}, \mathbf{Af}\} \quad (2)$$

Si \mathbf{Or}_i représente un organe quelconque dans le corps humain. L'ensemble des organes du corps est définit par :

$$\mathbf{Or} = \{\mathbf{Or}_1, \mathbf{Or}_2, \mathbf{Or}_3 \dots \mathbf{Or}_n\} \quad (3)$$

Soient \mathbf{Ir}_i une relation interne (fonction mathématique) pour un \mathbf{Or}_i qui utilise des valeurs de $\mathbf{SC} \in \mathbf{Or}_i$ et $\mathbf{SPar} \in \mathbf{Or}_i$. L'ensemble **Ir** est alors définit par :

$$\mathbf{Ir} = \{\mathbf{Ir}_1, \mathbf{Ir}_2, \mathbf{Ir}_3 \dots \mathbf{Ir}_n\} \quad (4)$$

Soient \mathbf{Er}_i une relation externe (fonction mathématique) entre deux ou plusieurs \mathbf{Or}_i qui utilise les valeurs de **C** et **Par**. L'ensemble **Er** est définit par :

$$\mathbf{Er} = \{\mathbf{Er}_1, \mathbf{Er}_2, \mathbf{Er}_3 \dots \mathbf{Er}_n\} \quad (5)$$

Soit $SName$ le nom du système de corps (comme le système digestif), les variables SC et SPV sont liées à un S_i concerné. Notant que, l'intersection entre SPV des différents systèmes n'est pas nécessairement vide, parce que certaines variables physiologiques peuvent affecter plusieurs systèmes, donc elles sont des variables communes entre ces systèmes. L'ensemble S est défini comme:

$$S = \{S_1, S_2, S_3 \dots S_n\} / S_i (SName, SC, SPV, Ir, Er) = \bigcup_{j=1}^n Or_{ij} \quad (6)$$

Soient Par_i les paramètres d'entrée pour le système, $Par = U_{SPar}$, $SPar \in S_i$. L'ensemble Par est défini par :

$$Par = \{Par_1, Par_2, Par_3 \dots Par_n\} \quad (7)$$

Soient C_i des constantes prédéfinies pour le système / $C = U_{SC}$, $SC \in S_i$. L'ensemble C est défini par :

$$C = \{C_1, C_2, C_3 \dots C_n\} \quad (8)$$

Soient PV_i les variables physiologiques contrôlés par le système / $PV = U_{SPV}$, $SPV \in S_i$. On définit PV par :

$$PV = \{PV_1, PV_2, PV_3 \dots PV_n\} \quad (9)$$

Enfin $Mp = f(Pf, Dm, S)$ serait un prototype global et flexible applicable à tout analyseur pour un patient et peut intégrer un ou plusieurs systèmes de l'organisme constituant un cas d'étude ou même tous les systèmes du corps. Ce prototype peut nourrir comme entrée le modèle de l'appareil avec ses valeurs nécessaires à travers les résultats des ensembles de relations (Ir et Er) ainsi que les variables physiologiques présenté par l'ensemble PV .

Exemple :

Le modèle qui a été développé principalement par Roman Hovorka et est donc référencé comme le modèle de Hovorka [67, 68, 69]. Dans ce model, Q_1 et Q_2 représentent les masses de glucose dans les compartiments accessibles et non accessibles; F_{0I} est le glucose flux total non-insulino-dépendant corrigée pour la concentration de glucose ambiante et FR est la clairance rénale du glucose; G est la concentration de glucose mesurable; EGP représente la production endogène de glucose; x_1, x_2, x_3 représentent trois et les actions de l'insuline sur la cinétique du glucose; S_1 et S_2 sont une chaîne à deux compartiments qui représente

l'absorption de la voie sous-cutanée administrée courte durée d'action; I décrit la concentration plasmatique d'insuline; D_1 et D_2 sont une chaîne à deux compartiments qui représente la quantité de glucides à digestion. BW est le poids du patient. Voici une partie des équations du modèle Hovorka :

$$\frac{dQ_1(t)}{dt} = U_G(t) - F_{01,c} - F_R(t) - x_1(t)Q_1(t) + K_{12}Q_2(t) + EGP_0(1 - x_3(t))$$

$$\frac{dQ_2(t)}{dt} = x_1Q_1(t) - (K_{12} + x_2(t))Q_2(t)$$

$$\frac{dS_1(t)}{dt} = u(t) - \frac{S_1(t)}{T_S}$$

$$\frac{dS_2(t)}{dt} = \frac{S_1(t)}{T_S} - \frac{S_2(t)}{T_S}$$

$$D_1 = D_1 + h * \frac{dD_1(t)}{dt} \quad / \quad h \text{ est la constante de variation par rapport au temps}$$

$$D_2 = D_2 + h * \frac{dD_2(t)}{dt}$$

Le taux d'absorption de l'insuline, $U_I(t)$ (mU/min) peut être calculée :

$$U_I(t) = \frac{S_2(t)}{T_S}$$

La concentration d'insuline, $I(t)$ (mU/L), est trouvée en résolvant l'équation différentielle suivante:

$$\frac{dI(t)}{dt} = \frac{U_I(t)}{V_I} - K_e I(t)$$

x_1 , x_2 , et x_3 sont calculées en utilisant les trois équations différentielles suivantes, en fonction de la concentration de l'insuline plasmatique et des paramètres :

$$\frac{dx_1(t)}{dt} = -K_{a1}x_1(t) + K_{b1}I(t) \quad ; \quad k_{b1} = S_{IT} k_{a1}$$

$$\frac{dx_2(t)}{dt} = -K_{a2}x_2(t) + K_{b2}I(t) \quad ; \quad k_{b2} = S_{ID} k_{a2}$$

$$\frac{dx_3(t)}{dt} = -K_{a3}x_3(t) + K_{b3}I(t) \quad ; \quad k_{b3} = S_{IE} k_{a3}$$

Chapitre 2 : Simulation et évaluation de la QoS des systèmes médicaux

La simulation du système biomédical et de son interaction avec le corps humain repose comme indiquée sur la figure 3, sur deux parties interdépendantes : la première « Simulation des Patients », représente le comportement du patient. Celui-ci est simulé en utilisant un modèle mathématique représentant au mieux les mécanismes du système organique humain. Les nombreuses constantes et paramètres qui interviennent dans le modèle sont choisis pour simuler la physiologie et la pharmacologie du patient de façon à fournir des réponses en temps réel à tout traitement donné.

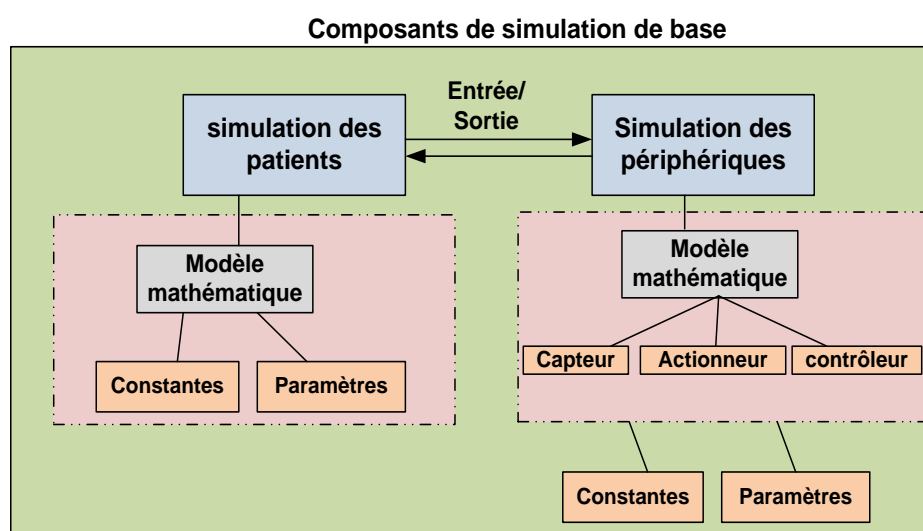


Fig. 3: Architecture du simulateur retenu

La deuxième « Simulation des périphérique » est utilisée pour simuler les dispositifs médicaux. Les modèles mathématiques utilisés offre aussi la possibilité de simuler les capteurs, les actionneurs et les contrôleurs associés. Cette approche permet d'approcher au mieux la fonctionnalité du dispositif médical afin de permettre une simulation complète du système. La finesse du modèle peut, on fonction des objectifs recherchés, être simple ne prenant en compte que les fonctionnalités apparente du dispositif pour une analyse grossière ou complexe allant jusqu'à prendre en compte des problématiques de mécanique ou de gestion de l'énergie pour une analyse approfondie. Les modèles mathématiques des patients et des périphériques associés à savoir les capteurs, les actionneurs et les contrôleurs ont été développés est programmés en utilisant la plateforme Keil [39, 40]. La simulation conjointe de ces deux modèles : simulation des patients et simulation des périphériques, permet

d'étudier le fonctionnement et la qualité de service de différents équipements biomédicaux face à différents scénarios.

2.1. Architecture du simulateur mis en œuvre

Un simulateur pour le domaine médical est un dispositif qui permet de reproduire, dans des conditions donnés de test, des phénomènes susceptibles de se produire dans le monde réel [41]. On associe souvent à un simulateur pour le domaine médical les scénarios de tests, les patients concernés et les dispositifs médicaux utilisés. Comme le montre la figure 4, le simulateur mis en œuvre pour notre recherche, englobe l'ensemble des éléments précédemment cités.

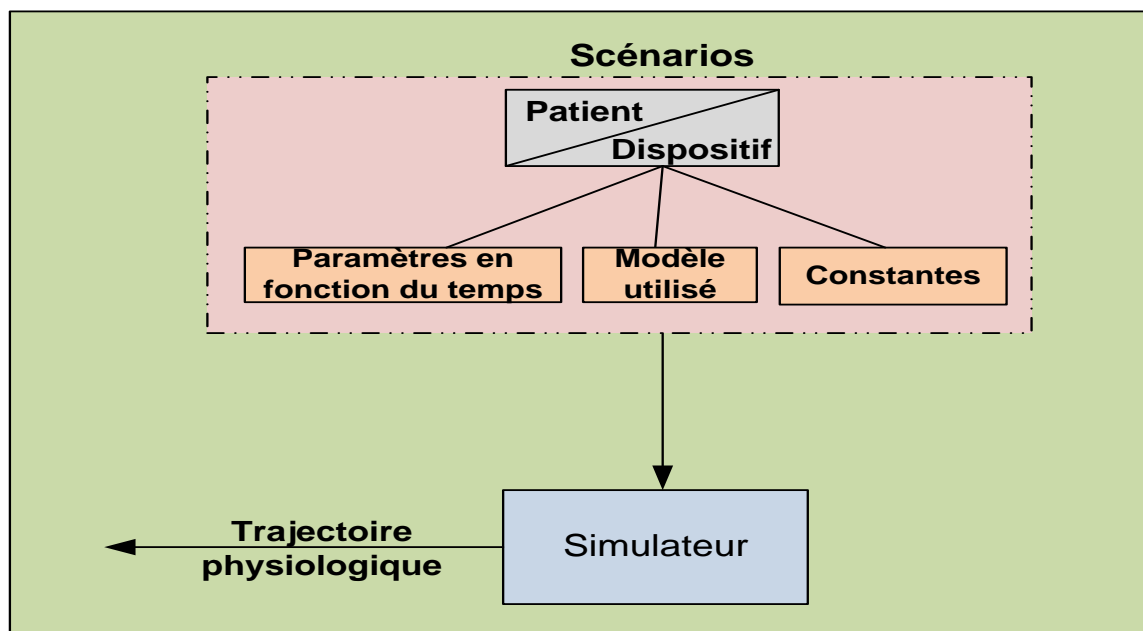


Fig. 4: Simulateur

Pour le patient, comme pour le dispositif à étudier, notre simulateur utilise des modèles mathématiques à base d'équations différentielles pour représenter le fonctionnement du dispositif médical et le comportement du patient. Dans le cas du patient, la physiologie peut évoluer en fonction d'un ensemble de paramètres et de constantes spécifiques au patient considéré. Les repas, leurs contenus et leurs périodicités peut par exemple être pris en compte par le modèle. De même, le model mathématique du dispositif médical permet de simuler les anomalies de fonctionnement de ce dernier. Le modèle permet en effet de modifier les constantes de configurations et les paramètres spécifiques du dispositif considéré. On peut par exemple simuler une mauvaise utilisation du dispositif ou un changement de sa configuration et observer la conséquence sur la trajectoire d'une ou des variables physiologiques du patient.

On peut aussi changer les scénarios de simulations on les rapprochant de cas cliniques connus et observer le comportement du dispositif [42] et son incidence sur la trajectoire physiologique du patient, etc.

L'analyse des résultats permettra ainsi d'améliorer les diagnostics et d'optimiser la performance des dispositifs considérés pour minimiser les risques sur les patients.

A partir des différents et nombreux scénarios choisis, des trajectoires physiologiques sont donc générées par le simulateur (Figure 5). L'analyse de l'ensemble des trajectoires permet, par comparaison à des trajectoires connues, de savoir s'il y a anomalie ou pas et par la même occasion de déduire une indication sur la qualité du service du système considéré. L'objectif étant de vérifier si le du système est capable de garantir un niveau de qualité de service (QoS) compris dans un intervalle de valeurs considérées satisfaisantes.

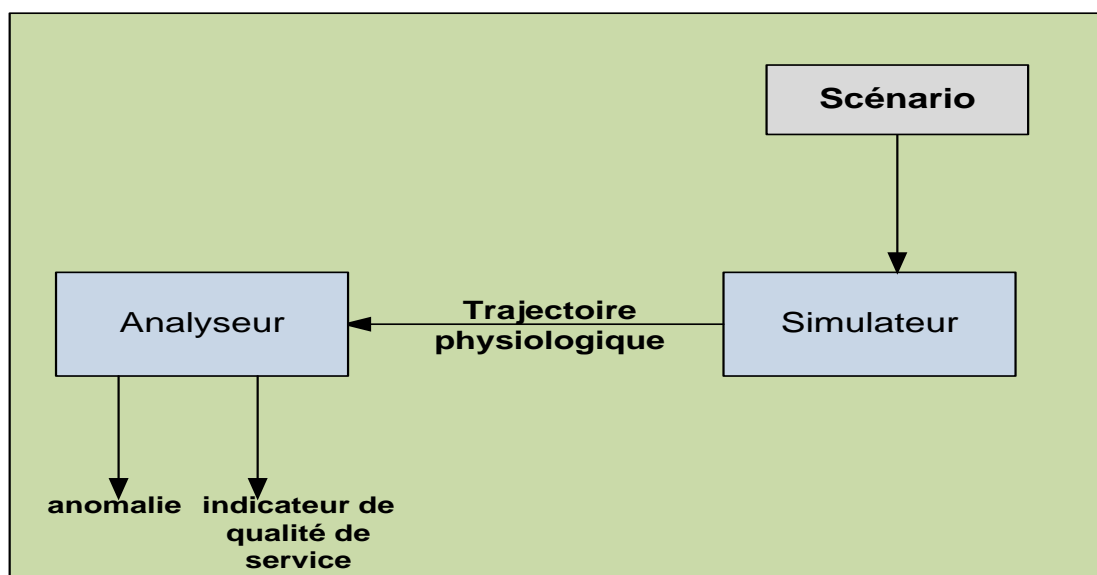


Fig. 5: Principe de l'analyseur

2.2. La qualité de service (QoS)

La qualité de service (QoS) est un indicateur qui peut couvrir différentes performances d'un système, elle porte principalement sur la disponibilité, la fiabilité, la sécurité et le coût. C'est une notion qui est beaucoup mis en avant pour qualifier les performances d'un système médical. Qu'il soit un service comme par exemple l'accès aux données d'un patient ou un dispositif comme par exemple un pacemaker [43, 44, 45, 46]. Dans le cas des dispositifs embarqués biomédicaux, maintenir une qualité de service à un niveau donné, c'est maintenir un paramètre physiologique à une valeur donnée, supposée normale. [La valeur normale d'un paramètre physiologique d'un patient correspond à la valeur qui doit rester inchangée ou avec de très légères variations de jour en jour, voire d'année en année]. Comme rappelé dans

l'introduction, dans tous les cas et quel-que-soit l'individu, la valeur normal d'un paramètre physiologique, doit être contrôlée est maintenue entre un niveau maximum et un niveau minimum spécifiques à l'organisme de l'individu. Un corps humain sain dispose de ses propres mécanismes de contrôle automatique. Basées sur le principe de la rétroaction, ces mécanismes de régulations maintiennent les paramètres physiologiques à des valeurs normales permettant à l'individu de vivre correctement, c'est cette régulation que nous essayons de reproduire par notre simulateur. L'évolution dans le temps des variables physiologiques étant contrôlées est décrite par des équations différentielles, lorsqu'un événement quelconque provoque le changement de l'état d'une variable, des rétroactions sont déclenchées pour compenser les variations afin de ramener le paramètre physiologique à sa valeur normal. La figure 2 illustre les rôles du capteur, du contrôleur et de l'actionneur dans la production de la réponse de contre-réaction. Les paramètres de contrôle de la variable physiologique peuvent se référer à des valeurs statiques ou à des valeurs dynamiques variables dans le temps. L'étude de chaque dispositif permet d'identifier les variables qui doivent être manipulées et comment elles doivent être modifiées afin d'atteindre les spécifications désirées. Si α_i représente l'ensemble des valeurs qu'un paramètre physiologique PV peut prendre depuis sa valeur normale à sa valeur maximale que nous noterons α_1 . De même, si β_i représente l'ensemble des valeurs que ce même paramètre physiologique peut prendre depuis sa valeur normale jusqu'à sa valeur minimale que nous noterons β_1 , on peut pour les valeurs possibles de PV écrire :

$$\{\beta_1, \beta_2, \beta_3, \dots, \beta_n\} < PV < \{\alpha_n, \dots, \alpha_3, \alpha_2, \alpha_1\} \quad (10)$$

Le suivi et l'analyse de l'évolution des valeurs prises par le paramètre physiologique PV peut être utilisé pour qualifier les performances d'un dispositif médical donné et ainsi mesurer sa qualité de service. On peut par exemple tolérer une valeur β_i ou une valeur α_i mais uniquement si cette dernière ne dure qu'un laps de temps supposé autorisé. La surface de la courbe entre t_1 et t_2 doit dans ce cas rester inférieure à une certaine valeur.

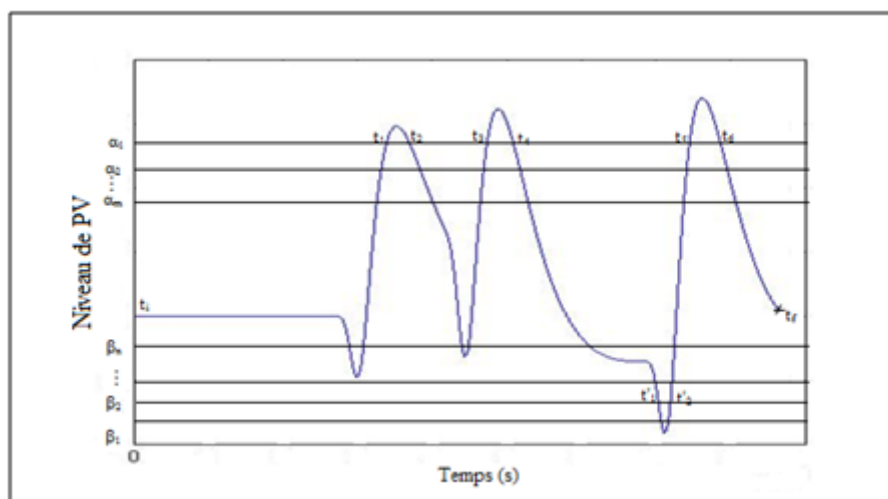


Fig. 6 : Représentation de l'indice de gravité

2.3. Quantification de la qualité de service

Comme indiqué par la figure 7, pour chaque scénario une indication sur la qualité de service du système est extraite.

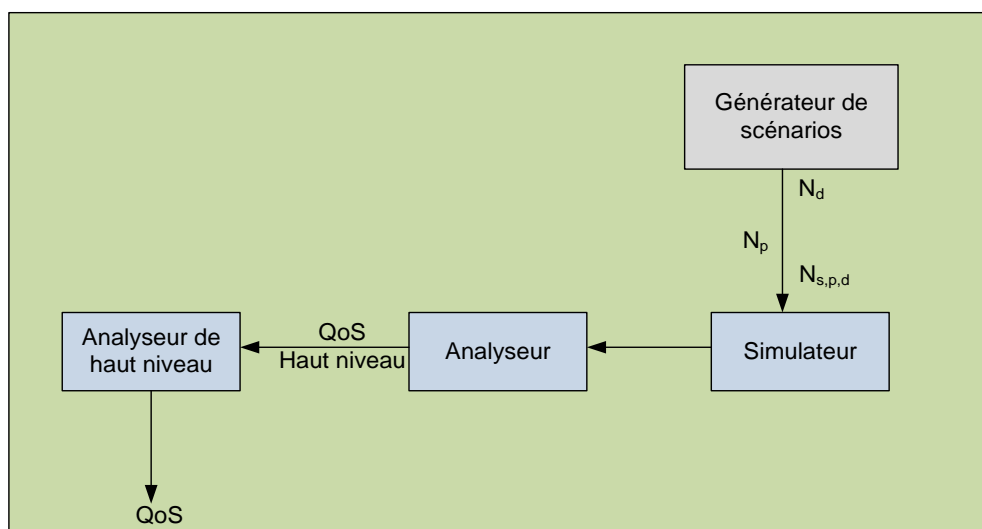


Fig. 7: QoS indicateur schéma

Après la création et la simulation de quelques millions de scénarios représentatifs d'une population importante et de cas pathologiques divers et variés, une indication sur la qualité de service globale du dispositif peut être récupérée. Dans le cadre de notre thèse nous avons mis en œuvre un système (Figure 7) complet capable de générer automatiquement les multiples scénarios nécessaires à l'alimentation du simulateur. Nous pouvons pour différents scénarios, modifier le nombre de dispositifs et leurs caractéristiques (N_d), le nombre de patients et leurs caractéristiques (N_p), nous pouvons aussi agir en même temps sur les patients, les dispositifs et les paramètres ($N_{s,p,d}$), etc. La création et l'utilisation de plusieurs scénarios permet par

simulation et analyse des résultats d'identifier les paramètres les plus pertinents. Afin d'arriver à calculer la bonne qualité de service (QoS), nous avons besoin pour créer les nombreux scénarios de beaucoup de connaissances sur la population ainsi qu'un certain nombre de voies génériques de trajectoires physiologiques.

Dans la figure 8, la partie représentée avec cadre pointillé visualise ce qui se passe dans la simulation d'un système pour calculer une valeur de QoS. On a une population virtuelle d'un ensemble de cas, accompagnés par des ensembles de n paramètres physiologiques représentant les n cas virtuelles. Chaque cas présente des caractéristiques représentées sous forme vectorielle par des paramètres physiologiques fixes et d'autres variables spécifiques aux patients. L'implémentation de ce vecteur sera traitée dans le prochain chapitre. Prenons le cas pour d'un patient virtuel atteint du diabète de type 1. Ce dernier sera représenté par un modèle de régulation du glucose et d'un certain nombre de paramètres spécifiques au patient.

Afin d'évaluer la qualité de service d'un dispositif médical donné pour la régulation du diabète du patient considéré, on injecte dans le modèle d'une population de référence la sortie du système à évaluer. La population est considérée comme une entrée pour le dispositif (actionneur, capteur et contrôleur) et permet de calculer, comme indiqué par la figure 8, la valeur de Y considérée comme une valeur de sortie approximative représentant la valeur de la QoS du système. Pour tout système qui possédant des paramètres d'entrées et des valeurs de sortie, nous pouvons par ce principe calculer par simulation une valeur QoS relative à une population et des périphériques associés.

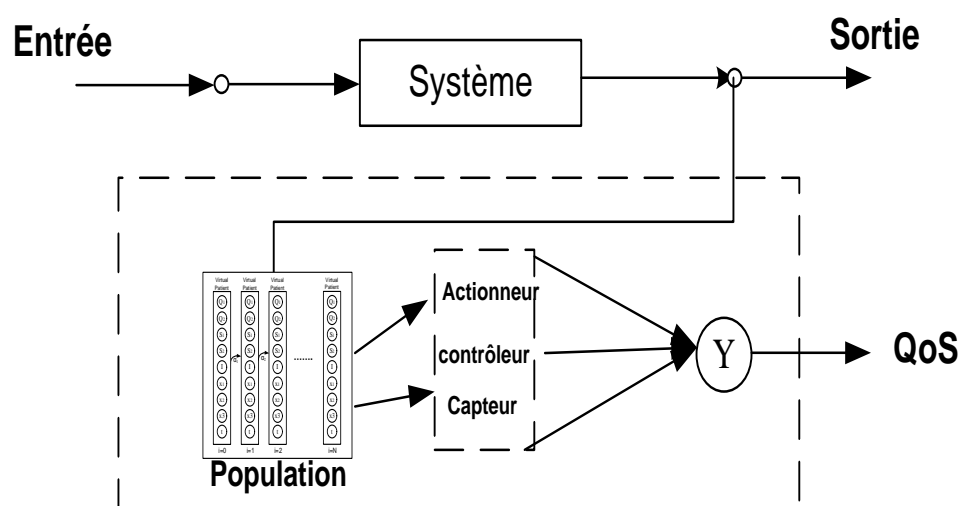


Fig. 8: QoS entrée et sortie

Le modèle en boucle fermée fonctionne comme une file d'attente. Générés automatiquement par le générateur de patients virtuels mis en œuvre, le comportement des

patients et du dispositif médical sont simulés dans différentes configurations. La figure 9 illustre ce fonctionnement. Cette figure peut être composée en trois sections. La première concernant la population des patients virtuels qui prend comme entrée Dm (repas) et un ensemble de paramètres et de constantes variables spécifiques au patient qui varient en fonction du temps t . Un aperçu de haut niveau du système permet de créer de nombreux cas cliniques qui peuvent bénéficier de systèmes à boucle fermée. La deuxième section représente une boucle fermée concernant la variable PV à contrôlée, le system S en charge, les résultats et le testeur du system pour analyser la performance du système. La variable PV à contrôler et le testeur sont interconnectés pour former un système de régulation en boucle fermée physiologique.

La rétroaction est nécessaire dans le cas d'une boucle fermée représentée comme troisième section.

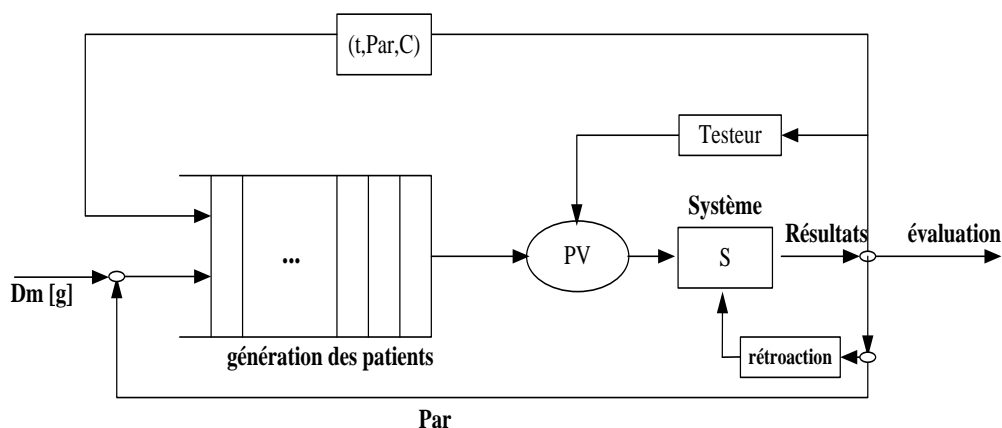


Fig. 9: Modèle du système en boucle fermée

- S : représente le système à contrôler.
- PV : représente la variable que le système essaie de contrôler.
- **Rétroaction** : aide à surveiller et à déclarer les valeurs.
- **Testeur** : représente notre modèle de test.
- (t, Par, C) : représentent un ensemble de paramètres et de constantes variables spécifiques au patient en fonction du temps t .
- Dm : représente les repas quotidiens pris par le patient.

2.4. Evaluation du QoS

Nous utilisons deux techniques pour évaluer la QoS offerte par le dispositif à étudier. La première technique calcule un indice de gravité IG_H (g) indiquant la gravité de la valeur mesurée sur l'état du patient. Cet indice est calculé sur un intervalle de temps $[t_i \ t_f]$, comme indiqué sur la figure 10, l'évolution d'une fonction $f(t)$ représentative de l'état d'une variable physiologique à contrôlée.

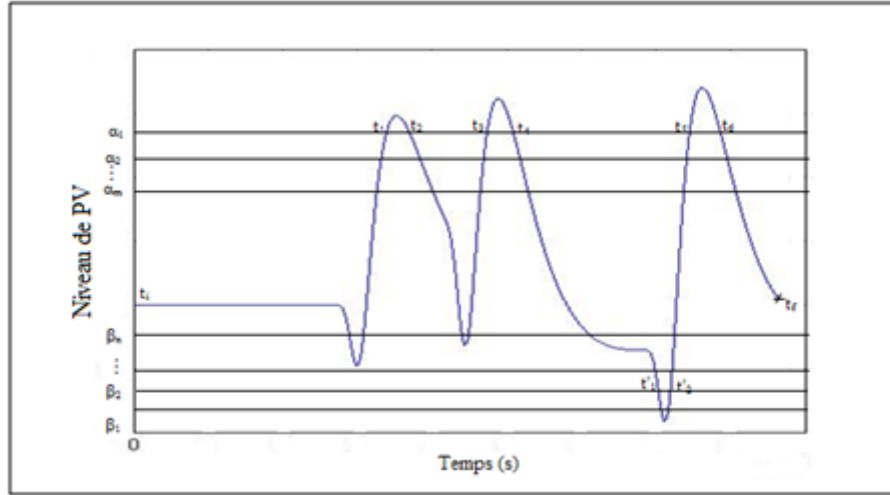


Fig.10: Évolution du niveau d'un PV en fonction du temps

Considérons $f(t)$ la fonction représentant le niveau du PV déjà contrôlé sur un intervalle de temps $[t_i \ t_f]$. Les paramètres α_m et β_m sont des constantes prédéfinies avec $\alpha_m = \{\alpha_1, \alpha_2, \alpha_3 \dots \alpha_n\}$ et $\beta_m = \{\beta_1, \beta_2, \beta_3 \dots \beta_n\}$. Soit T_{Hyper} la durée en $[t_i \ t_f]$ dans le cas de la valeur maximale et la durée T_{Hypo} dans $[t_i \ t_f]$ dans le cas de la valeur minimale.

Pour chaque $t_j / f'(t_j) = 0$ et $f(t_j) > \alpha_m$:

$$\exists t_K, t_L (t_K < t_j < t_L) / f(t_K) = f(t_L) = \alpha_m \text{ et } T_{Hyper} = \sum (t_L - t_K) \quad (11)$$

Pour chaque $t'_j / f'(t'_j) = 0$ et $f(t'_j) < \beta_m$:

$$\exists t'_K, t'_L (t'_K < t'_j < t'_L) / f(t'_K) = f(t'_L) = \beta_m \text{ et } T_{Hypo} = \sum (t'_L - t'_K) \quad (12)$$

Ensuite, pour une simulation donnée, il existe une valeur $g = \int_{t_k}^{t_f} f(t)$, qui représente la surface délimitée par la courbe et la droite α_m ou β_m . Cette surface définit un indice de gravité IG_H (g), qui indique la gravité du cas du patient. Cette surface peut être calculée par :

$$IG_H(g) = \text{trapz}(x_1, f') - \min(f')(\max(x_1) - \min(x_1)) \quad (13)$$

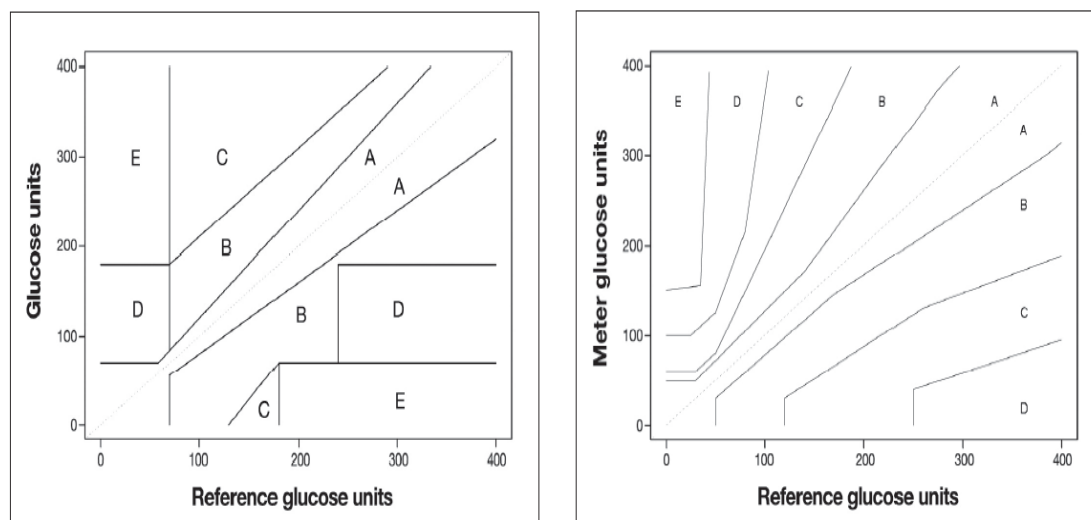
La deuxième, basée sur une grille d'analyse d'erreurs améliorée, relativise la dangerosité de la valeur mesurée en ajoutant un critère de temps. Pour le cas du diabète par exemple, nous avons pris comme intervalle de temps les 24 heures d'une journée et d'une nuit que nous avons répartis sur trois créneaux : autour du petit déjeuner, autour du déjeuner et autour du dîner. La moyenne des valeurs du paramètre physiologique mesurées sur un créneau de temps représente ainsi un point de la grille d'analyse. L'objectif dans les deux cas étant de mesurer les différences entre les valeurs mesurées et les valeurs normales afin d'observer les réactions du dispositif médical et ainsi qualifier ses performances.

Représentation graphique de la grille d'analyse d'erreur :

Le principe d'évaluation des performances d'un dispositif électronique par la mesure du paramètre à évaluer et sa comparaison à une grille d'erreur de référence connue est beaucoup utilisé dans le domaine des dispositifs biomédicaux [49, 50]. C'est par exemple le cas pour la mesure des performances des dispositifs de dépistage du diabète ou de contrôle de glycémie. Pour le cas du diabète, les grilles d'erreurs de mesures les plus connues pour estimer les performances de mesures des capteurs de contrôle de glycémie sont celles proposées par Clarke et Parkes [50, 51]. Clarke et Parkes ont conçu des grilles d'analyse d'erreurs (EGA : Error Grid Analysis) en prenant en considération non seulement la différence entre la valeur de glucose mesurée par le dispositif et la valeur de glucose de référence dans le sang mais en intégrant aussi la signification clinique de cette différence à savoir la dangerosité de la mesure. Comme indiqué sur la figure 11, cinq catégories de risques ont été définies :

- **A** : fonctionnement normal
- **B** : peu ou pas d'effet sur les résultats cliniques.
- **C** : de nature à affecter les résultats cliniques.
- **D** : peut avoir un risque médical important.
- **E** : peut avoir des conséquences dangereuses.

L'abscisse X représente la valeur de glucose de référence, donc la valeur acceptable et l'ordonnée Y représente la valeur de glucose mesurée. Si par exemple la valeur de référence vaut 300 et la valeur mesurée vaut 300, le risque se trouve dans la catégorie A considérée comme normal donc pas de danger. La grille de Parkes permet d'éviter les ambiguïtés en supprimant les zones de conflit qu'on peut observer dans la représentation de Clarkes ; exemple entre la zone D et la zone A où le risque d'une mauvaise interprétation est possible.



Clarke

Parkes

Fig 11 : Grilles d'analyse d'erreur de Clarke et Parkes

Nous commençons à partir de l'indice de la représentation de testeur de gravité, les différents simulations faites et en utilisant une fonction de normalisation, nous pouvons représenter la simulation comme un point dans le but de construire une nouvelle grille d'erreur et l'utiliser pour évaluer la précision des mesures de niveau de glucose faites par les patients. Nous devons trouver un moyen de prendre les valeurs d'une courbe d'une simulation, représentée par un vecteur, qui sont sur la plupart des scénarios autour de 80 935, et les tracer comme un point dans la grille. Ce vecteur ressemblera à celui-ci:

Str = [90.374962 90.381195 90.387421 90.393639]

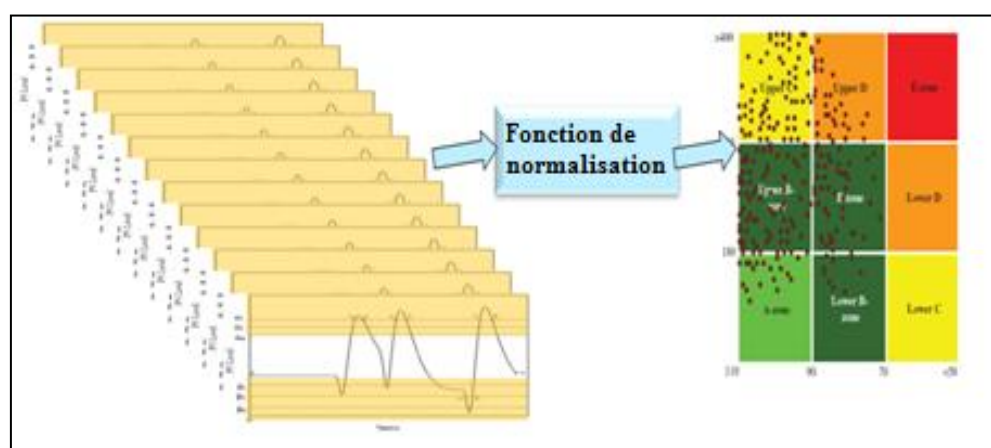


Fig. 12: Représentation graphique en utilisant la fonction de normalisation

La simulation peut être représentée comme un point afin de construire une nouvelle grille d'erreur [49, 50, 51, 52, 53] et de l'utiliser pour évaluer l'exactitude des mesures du niveau de

glucose faites par les patients (Figure 12). En résumé, chaque simulation est représentée par un point dans la grille; ce point est calculé en utilisant une fonction de normalisation. Ayant de nombreuses simulations de scénarios aident à tracer ces points dans cette grille. Le terme percentile est souvent utilisé dans la déclaration des résultats des tests, on l'a représenté sous forme d'équations (Appendice C).

Cette fonction de normalisation peut être percentile ou une unité de mesure qui aide à calculer un point (par exemple médiane). Le 2.5th et 97.5th percentiles représentent un sens unique entre deux boîtes de test dans le meilleur et le pire des cas, respectivement, de même que la médiane (50th percentile) indiquerait une valeur normale entre deux points de mesure. Si vous avez un ensemble de valeurs, celles de la "percentile 97,5th" sont les valeurs supérieures à 97,5% des autres.

La fonction `prctile (vect, par)` (tableau 1) prennent deux paramètres, la première est les valeurs du vecteur de la simulation et la valeur percentile comme second paramètre, par exemple 2,5.

```
prct25p1 = prctile(str,per)
```

Une partie de ces fonctions (la percentile, la moyenne, l'écart-type, l'erreur-type et la médiane.), de même les algorithmes sont décrits ci-dessous (l'annexe C). Tableau 1 représente l'algorithme utilisé pour calculer la fonction de percentile.

Algorithme : Algorithme de calcul pour la fonction percentile	
1.	Soit <i>len</i> la longueur du vecteur de données triées qui représentent les points de simulation and $0 < p \leq 100$ la valeur du percentile.
2.	Si $p > 100$ or $p < 0$ alors retourner les éléments du vecteur unique sinon
3.	Calculer le percentile estimé: $position = (len + 1) * p / 100$
4.	Poser $n = p / 100 * (len - 1) + 1$;
5.	Soit <i>left</i> l'élément en position $\text{floor}(position)$ dans le vecteur et soit <i>right</i> l'élément suivant in the vecteur, où <i>floor</i> est le plus grand entier inférieur ou égal à la position spécifiée.
6.	Si $position \geq 1$ alors calcule les valeurs de <i>left</i> et <i>right</i> . sinon retourner <i>left</i> comme le premier élément du vecteur et <i>right</i> comme le second.
7.	Si <i>left</i> égal <i>right</i> alors retourner la valeur du <i>left</i> . sinon retourner $left + (n - \text{floor}(n)) * (right - left)$

Tableau 1: percentile algorithme

La figure 13 illustre le modèle globale incluant le modèle du patient et du dispositif médical à évaluer retenu pour notre étude.

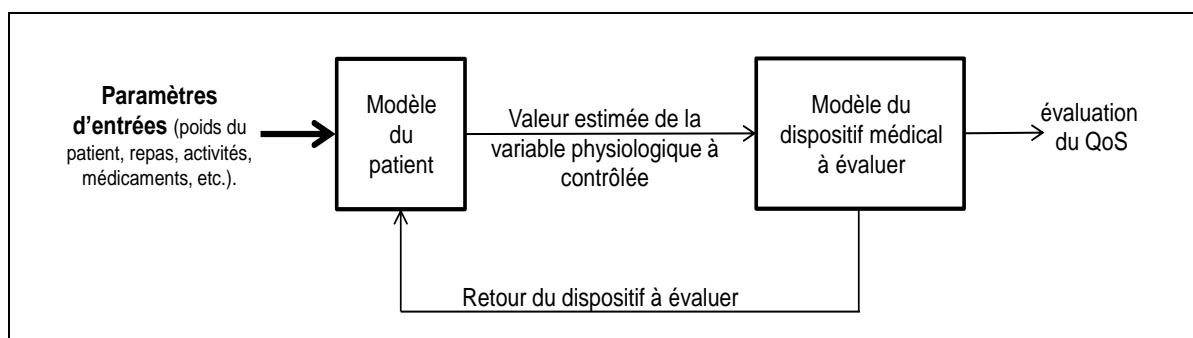


Fig. 13 : Schéma de principe d'évaluation du dispositif

Pour une pathologie donnée, le modèle du patient permet, à partir de paramètres d'entrées spécifiques à un patient (poids, activités sportives, repas, etc.), d'estimer une valeur de la variable physiologique à surveillée. Le modèle du dispositif médical à évaluer analyse la valeur estimée et se charge de la correction nécessaire à apporter pour approcher au mieux une valeur normale, c'est-à-dire une valeur dans les limites tolérées par l'organisme du patient. Pour le cas du diabète par exemple, le retour peut être une consigne d'injection automatique ou pas d'une dose d'insuline. La figure 14 illustre comment fonctionne le modèle du dispositif médical à évaluer. En suivant les retours du dispositif médical et leurs impacts sur l'évolution dans le temps de la variable physiologique à contrôlée, on peut quantifier la qualité de service du dispositif.

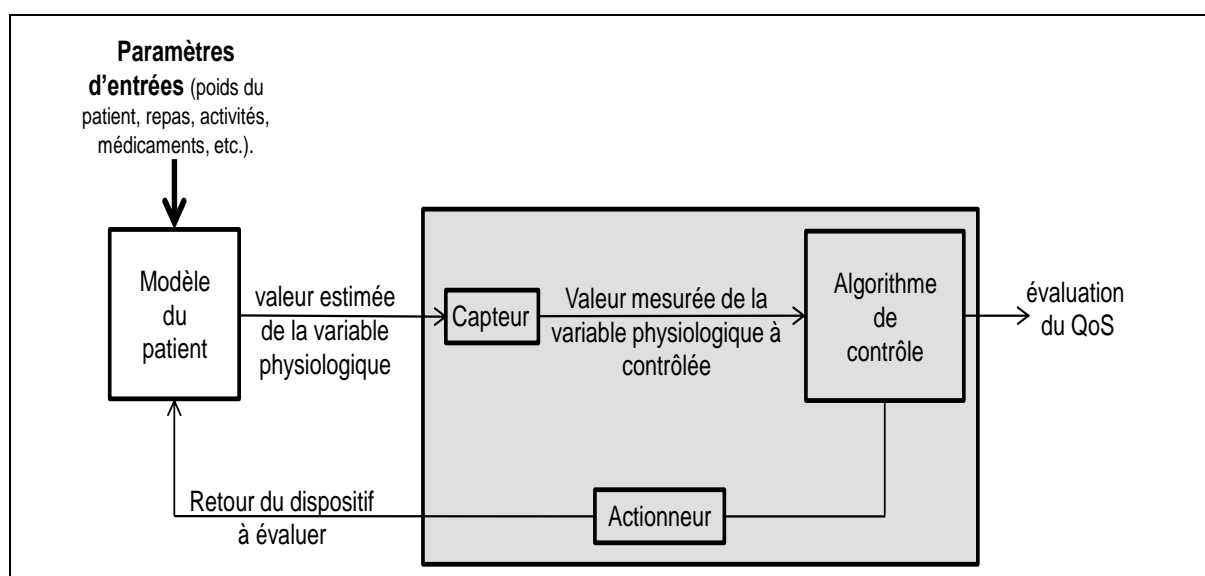


Fig. 14 : Modèle de régulation d'une variable physiologique

L'analyse de la performance des dispositifs biomédicaux aide à identifier l'utilisation des capteurs, des actionneurs ou des contrôleurs. Dans ce travail, la représentation de la grille permettra non seulement de simuler les performances des capteurs actionneurs mais aussi et contrôleurs (figure 15). Analyser la performance des dispositifs biomédicaux aide à identifier l'utilisation de ces capteurs, actionneurs ou contrôleurs. En résumé, en utilisant la simulation des composants suivants: capteurs, actionneurs ou de contrôleur, et une fonction de normalisation, nous pouvons représenter cette simulation en utilisant la grille d'analyse.

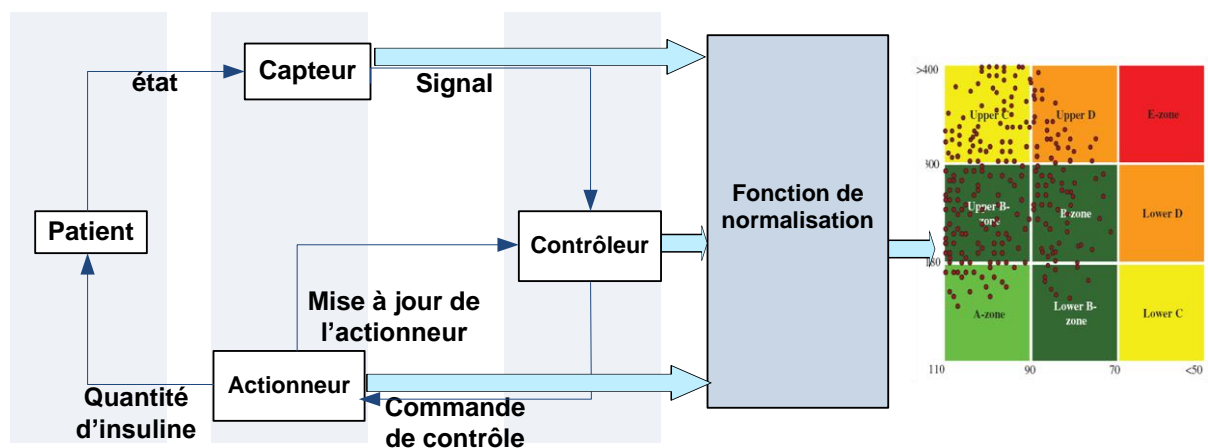


Fig. 15: Simulation des performances d'un équipement

Chapitre 3 : Implémentation et étude cas

Les avancées en recherche dans les domaines de l'électronique et de l'informatique embarqués ont permis une forte introduction des dispositifs de diagnostic et de suivi pour des applications biomédicales [54]. Pour le cas du diabète par exemple, il est nécessaire de comprendre l'effet de l'insuline et des hydrates de carbone sur l'évolution de la glycémie d'un patient spécifique. De nombreuses équations et modèles mathématiques ont été utilisés pour créer des simulateurs pour tester différents types de traitement et ont permis d'explorer nombreuses approches de contrôle et de régulation automatique de la glycémie [57].

Dans ce chapitre, nous présentons la mise en œuvre du modèle mathématique développé pour simuler la régulation du glucose dans le sang d'un patient. Le fonctionnement et le suivant : un patient avale plusieurs bonbons sucrés, le système digestif les transforme, ce qui provoque une augmentation rapide de la concentration sanguine. L'augmentation du niveau de glucose stimule les cellules du pancréas responsables de la fabrication de l'insuline, qui libèrent alors dans le sang. L'insuline accélère l'absorption du glucose par la plupart des cellules et favorise son stockage sous forme de glycogène dans le foie et les muscles; le corps en quelque sorte mis en glucose en réserve. Par conséquent, les rendements de glucose à la valeur de référence normale et l'événement qui a déclenché la sécrétion d'insuline diminuent également. Glucagon, une autre hormone pancréatique, a un effet opposé. Il est libéré lorsque les taux de glucose sont inférieurs à la valeur de référence.

Nous commençons par le système d'insuline glucose et nous créerons un nouveau Framework afin de pouvoir tester les performances de différents composants du système. Le Framework consiste à simuler un modèle mathématique du corps humain, de le mettre en œuvre dans un microcontrôleur, de développer un algorithme de contrôle paramétriques du modèle afin de montrer comment les dispositifs médicaux peuvent être reliés entre eux pour former un système en boucle fermée physiologique. Ce chapitre traite de la simulation mise en place, l'explication du travail de modélisation et la présentation de l'algorithme de contrôle.

Nous présentons également les trois modèles qui sont mis en œuvre : le modèle de corps humain, le modèle de pancréas artificiel et le modèle des capteurs et des actionneurs. Nous proposons un nouveau modèle pour la simulation globale des équipements biomédicaux (y compris les modèles d'interaction humaine). Nous commençons par une étude *in silico* pour les patients diabétiques de type 1 sucré à l'aide d'un modèle mathématique, la mise en œuvre d'un algorithme de contrôle. Ce modèle a été conçu pour fonctionner dans la boucle fermée de

l'insuline au glucose. Ensuite, le modèle de pancréas artificiel a été mis en œuvre pour commander le modèle mathématique du corps humain.

3.1. Système de glucose-insuline

Des relations existent entre les différents organes du corps humain : le foie, les muscles, le pancréas... afin de formuler un système de glucose-insuline, et ajuster la concentration du niveau de glucose dans le sang (figure 16). Le système de glucose-insuline au sein du corps humain agit normalement comme un régulateur de la concentration du glucose dans le sang (BG), de ce fait empêchant la hausse anormale de la glycémie (hyperglycémie), ou le niveau de glucose est anormalement basse (hypoglycémie). Le système de glucose-insuline est un exemple d'un système physiologique en circuit fermé. Le règlement normal du niveau de glucose sanguin est réalisé par le système de glucose-insuline. Une personne en bonne santé a normalement un taux de sucre entre 70-110 mg/dl.

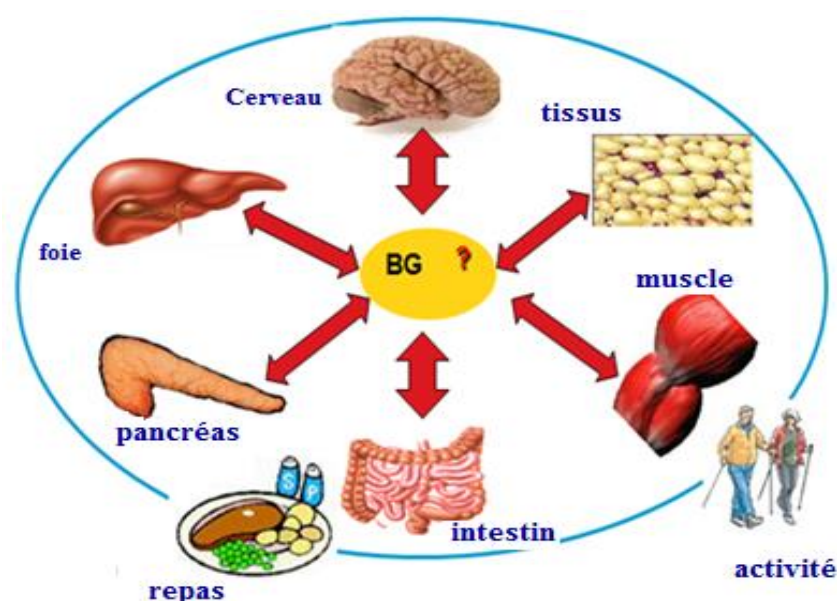


Fig. 16: Le système de glucose-insuline.

Le niveau de glucose sanguin devrait être maintenu dans une gamme très étroite ; l'insuline et le glucagon, sécrétés du pancréas, sont les hormones qui règlent ce niveau. Quand le contrôle des niveaux d'insuline échoue, le diabète résultera. L'insuline est une hormone produite par les cellules β des îlots de Langerhans dans le pancréas. Un niveau élevé d'insuline favorise le stockage du glucose, et un niveau bas d'insuline signale le besoin de libération des carburants de glucose, actuellement stockés, de nouveau dans le flux

sanguin. Le glucagon et l'adrénaline signale le foie pour sortir le glucose. Trop d'enlèvement de glucose de la circulation sanguine peut avoir comme conséquence les niveaux dangereusement bas de glucose sanguin. Le glucagon et l'insuline font partie d'un système de rétroaction qui maintient le glucose sanguin au niveau correct. Par exemple en cas d'hypoglycémie, les cellules α réagissent en libérant le glucagon, qui agit sur les cellules de foie, les faisant décharger le glucose dans le sang jusqu'à ce que la personne soit de retour dans le cas normal.

Dans un tel système, il y a un besoin de comprendre l'effet de l'insuline et des hydrates de carbone sur l'évolution de glucose sanguin pour un patient spécifique. Ceci est fait par la considération du système de glucose-insuline, ses entrées, ses sorties et de le modéliser. Dans ce cas, nous emploierons ces paramètres pour examiner les dispositifs biomédicaux sans utilisation de vrais patients.

L'intégration entre le système humain et le système électronique à agir en tant qu'un seul système est montrée dans la figure 17. Le système de glucose-insuline avec un dispositif de surveillance qui prennent le niveau de glucose et envoient le signal au contrôleur, le contrôleur communique avec un dispositif d'actionnement en envoyant la commande de contrôle, c'est l'actionneur qui envoie la quantité d'insuline réglée par le contrôleur. En bref, ce système intégré se compose du patient, du capteur, du contrôleur et d'un actionneur. L'objectif est de simuler le système entier de cette co-simulation, en d'autres termes, d'établir un système complet avec la capacité de la simuler et de détecter la performance de toutes les composantes du système.

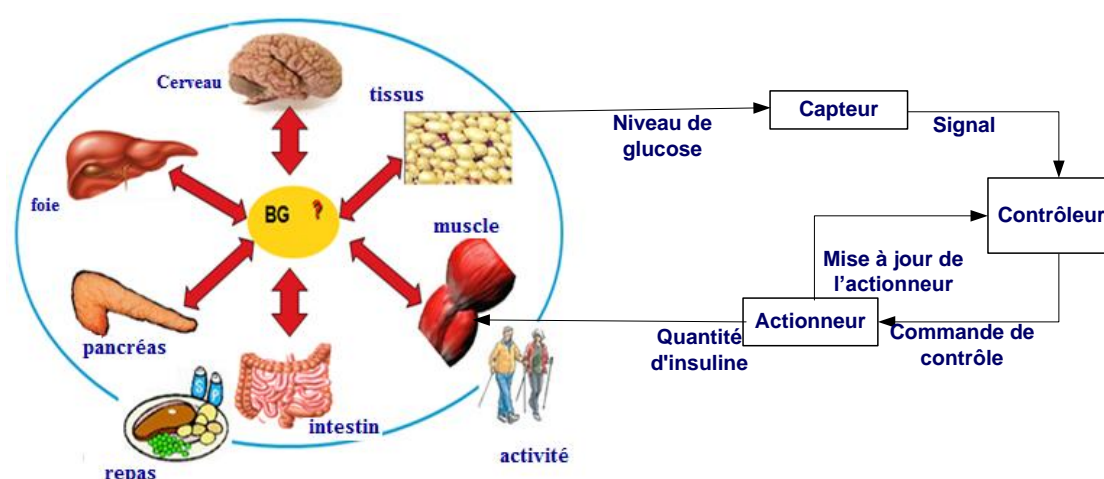


Fig. 17: Schéma d'interaction du système intégré.

L'objectif est de formuler une co-simulation des modèles de système physiologiques et systèmes embarqués (codes, capteurs, actionneurs...), et la représentation du règlement de la glycémie sous forme de modèles mathématiques. Les modèles mathématiques de la régulation du glucose ont été étudiés au cours des années. Récemment, la modélisation d'un système insuline-glucose a été présentée [71].

Tout d'abord, nous avons validé le modèle de corps humain en utilisant "Hovorka" que nous avons implémenté selon le scénario de référence définie dans [68, 69]. Nous l'avons programmé à l'aide du Keil en développement un programme de code intégré, qui a agit en tant que patient. Dans ce scénario, l'insuline est prise avant le repas. Nous avons considéré les repas prises par une personne, qui représentent le petit déjeuner, le déjeuner et le dîner dans une journée normale. Tout comme un point de référence, nous avons utilisé $\text{CHO}_1 = 45\text{g}$ et $\text{CHO}_2 = \text{CHO}_3 = 70\text{g}$, ce qui signifie que nous avons supposé que la quantité de glucides consommés pour le petit déjeuner est 45g, et ainsi de suite. Les doses d'insuline sont exactement les mêmes avec $2\text{U} / \text{L}$ pour le petit déjeuner et $3\text{U} / \text{L}$ pour le déjeuner et le dîner. Le graphique de la figure 18 représente ce cas, nous avons remarqué qu'il est lisse à l'intérieur de la limite en tout temps. Ce scénario correspond à l'injection d'insuline habituelle faite par un diabétique bien éduqué.

Nous avons simulé une vie quotidienne d'un patient en utilisant le modèle Hovorka, où le patient prend de l'insuline avant chaque repas. La progression de la simulation est considérée pour le cas d'une personne ayant un diabète qui tente de maintenir son niveau de glucose dans les limites. t_1

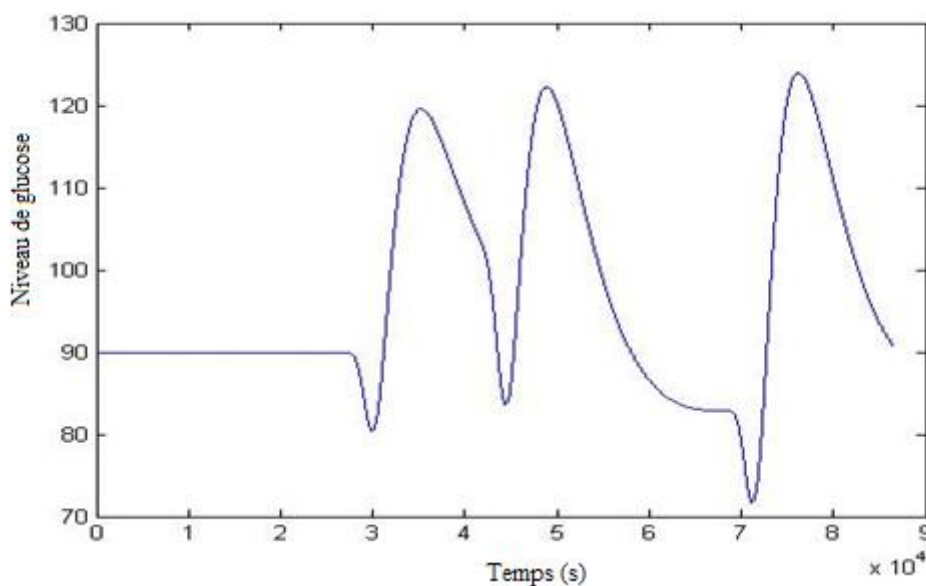


Fig. 18: Insuline administrée avant le repas

3.2. Implémentation d'un pancréas artificiel

La plupart des travaux proposés de la mise en œuvre de ce modèle sont effectués sous Matlab / Simulink ou logiciel java [72]. C'est souvent en raison de la complexité de l'utilisation du microcontrôleur et l'énorme quantité de temps nécessaire pour mettre en œuvre ces méthodes lors de l'incorporation des contraintes. L'objectif supplémentaire est de vérifier si la dose calculée est autorisée à être administrée. Les différents membres de la famille de microcontrôleurs 8051 conviennent à une vaste gamme de projets [73]. Dans cette partie, nous avons choisi de travailler avec un microcontrôleur 8051 afin de simuler notre système. Nous avons également simulé ce modèle en utilisant le programme Keil. Nous avons programmé le contrôleur en créant un programme *C* embarqué qui agissent comme pancréas artificiel. Son rôle principal était de réguler le niveau de sucre dans le sang par injection d'insuline (figure 19).

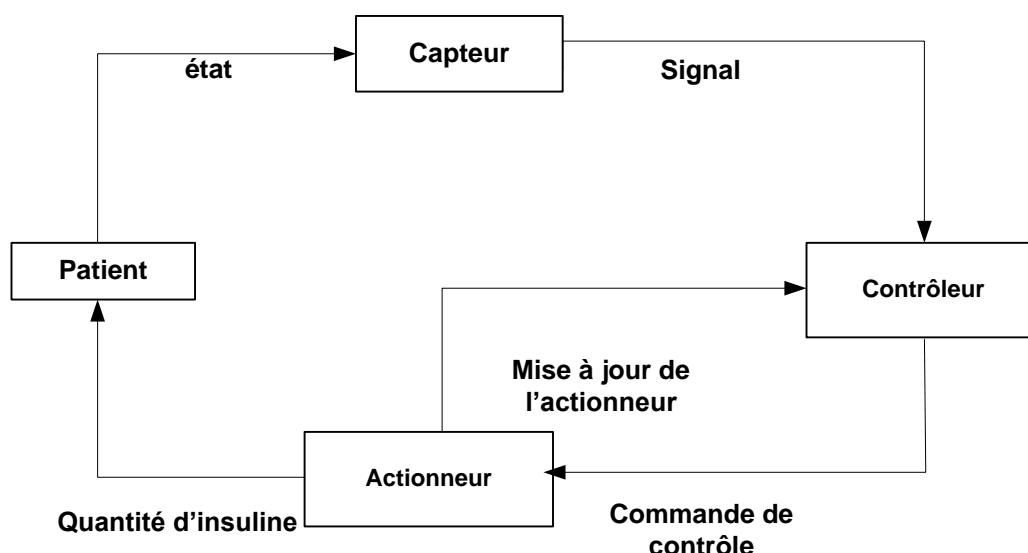


Fig. 19: Représentation en boucle fermée

Après que le capteur lit le niveau de glucose et envoie le résultat au contrôleur, ce dernier calcule la dose nécessaire pour maintenir un niveau existant dans les limites de glycémie entre 70 mg/dl et 110 mg/dl.

Le système mesure automatiquement le niveau de glucose dans le corps du patient. Les lectures consécutives, provenant du capteur intégré, sont comparées afin de fournir de l'insuline en cas de besoin. L'insuline est uniquement livrée dans les circonstances où il semble que le niveau de glucose est susceptible d'aller à l'extérieur de cette plage. La dose

proposée comme le glucagon, devient actif lorsque le niveau de glucose est inférieur à 60 mg/dl.

La configuration de la simulation est illustrée à la figure 20. Les entrées pour le patient virtuel sont le glucose et l'insuline, alors que la sortie $U(t)$ est la concentration de glucose dans le sang.

Un algorithme de prédiction parallèle a été utilisé pour calculer les futures valeurs de concentration de glucose dans le plasma envoyées au contrôleur. Les valeurs mesurées ou estimées du glucose dans le sang à partir du simulateur ont été utilisées par le contrôleur pour déterminer les doses d'insuline ou de glucagon à être administrée au patient virtuel.

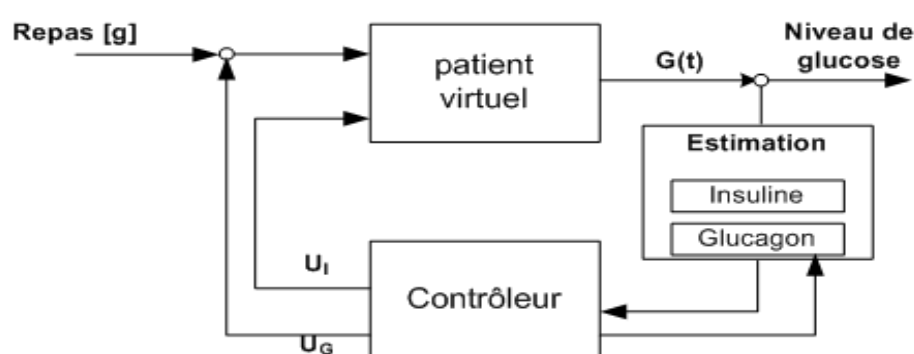


Fig. 20: Simulation boucle fermée

Le pancréas artificiel microcontrôleur (*APM*) choisit automatiquement la bonne dose à injecter en fonction du niveau de glucose en s'appuyant sur les règles de niveau de l'algorithme de contrôle de la glycémie. L'*APM* teste plusieurs fois le niveau de glucose dans le jour et se base sur son algorithme pour décider d'injecter ou non. Dans le cas d'injection, il calcule la dose nécessaire pour être injectée. Le tableau ci-dessous (tableau 2) décrit la règle de base qui a été principalement prise en compte pour construire notre algorithme de contrôle.

Règle#	Description
1	Faible niveau de sucre
2	Taux de sucre moyen
3	Haut niveau de sucre
4	Augmentation du niveau de sucre
5	Taux de sucre stable
6	Taux de sucre en chute
7	Taux de croissance est en baisse
8	Taux de croissance augmente
9	Taux de décroissance augmentant

10	Taux de décroissance diminuant
11	Administrer la dose calculée
12	Gestionnaire de dose quotidienne maximale

Tableau 2: Principales règles de l'algorithme de contrôle

La définition des règles de base pour la qualité du service, avec la capacité du système de répondre à des critères de décision médicaux. Cette qualité de service doit respecter les valeurs dans le tableau 3 ci-dessous.

Paramètres	Valeur	Description
SafeMin	70	Niveau minimum de sécurité de la glycémie
SafeMax	110	Niveau de sécurité maximale de la glycémie
MaxDailyDose	25	La dose maximale de l'insuline dans 24 h
MaxSingleDose	5	La dose maximale en une seule injection
MinDose	1	La dose minimum pour maintenir une tendance actuelle de la glycémie
MaxDose	4	La dose maximum pour maintenir une tendance existante de sucre dans le sang

Tableau 3: Les paramètres spécifiques

Les résultats sont présentés dans le graphique ci-dessous (figure 21.a) avec le même scénario pour le repas comme dans le premier modèle. Les résultats sont bien sûr pires que le cas précédent à la décision humaine. Nous pouvons voir que, après le petit déjeuner et le dîner, le glucose dans le sang est trop élevé (cf. tableau III.1). En effet, l'injection humaine a été décidée une heure avant le repas, l'APM ne peut détecter un changement de PV seulement après le repas, ce qui veut dire après le niveau de glucose est levé.

Notre objectif n'était pas de développer un pancréas artificiel idéal qui aura probablement d'anticiper le repas afin de réguler la glycémie dans le bon intervalle. Dans ce travail, le dispositif de commande n'a été utilisé que dans le but de tester les performances de tous les composants du système.

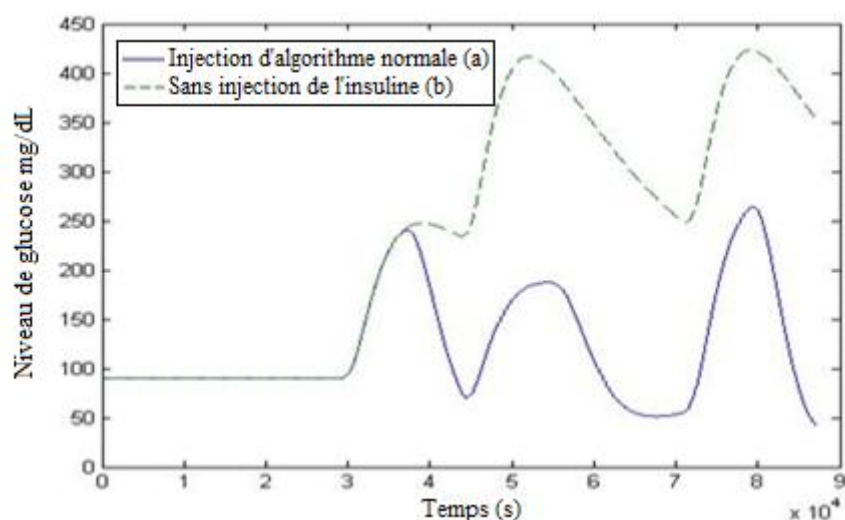


Figure 21: Avec ou sans injection d'insuline

Nous pouvons voir dans la figure 21.b ce qui se passera sans *APM*, la glycémie atteint une valeur très élevée après le déjeuner et le dîner. Afin de formuler si le testeur est efficace, nous avons changé quelques paramètres dans le système. Si un problème se produit au niveau d'actionnement, ceux-ci produisent une addition d'une ou deux unités ou plus dans la quantité de la dose injectée. Un autre cas est considéré, en cas de problème au niveau du capteur, le niveau de glucose mesuré est élevé, par exemple 10% ou 20% (Figure 22). Ceux-ci montrent comment le changement de la dose d'insuline injectée ou avoir un problème dans un niveau de matériel ou de logiciel peuvent affecter le système. Ces résultats permettent de tester le système, faire un bon changement de performance et de tester la réponse du système dans des nombreux cas.

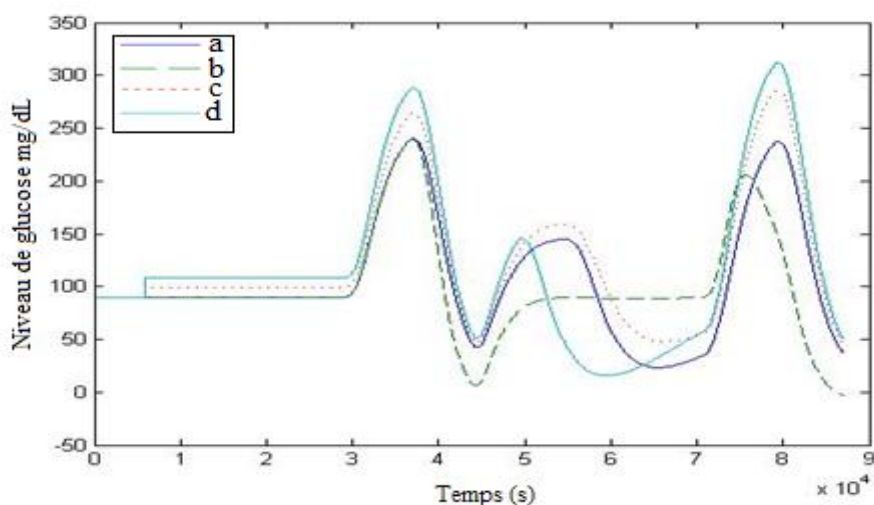


Figure 22: Simulations des résultats: (a) une unité injecté (b) deux unités injectées (c) + 10% sur le taux de glucose (d) + 20% sur le niveau de glucose

3.3. Etude in silico

Nous avons essayé de mettre en œuvre la technique in silico sur un patient virtuel ayant un diabète de type I (T1DM). L'expérimentation in silico [82, 83, 84] présente des nombreux avantages en fournissant une plus grande productivité du travail, un coût minimal et des simulations plus précises grâce à des modèles plus sophistiqués. Dans la partie simulation, nous avons utilisé un modèle mathématique "Hovorka" qui a été testé sur un procès in-silico des sujets diabétiques type I pendant 4 jours, recevant le petit déjeuner, le déjeuner et le dîner chaque jour. De plus, l'algorithme du contrôleur a été programmé en utilisant le langage C embarqué et il a été simulé à l'aide du cycle par Keil de cycle sur le type d'architecture 8051.

On a supposé que la simulation a été réalisée à partir de minuit avec un patient virtuel ayant un état spécifique constant. Le tableau 4 ci-dessous représente le patient virtuel dans les 4 jours de visite in-silico. La performance de boucle fermée de la glycémie est contrôlée lors de ces jours. Chaque jour, un hydrate de carbone étant servi par exemple dans le deuxième jour, un grand déjeuner de 100 [g] est consommé. La quantité de repas consommée et l'unité d'insuline injectée sont différentes d'un jour à l'autre. Nous avons essayé, par exemple, d'augmenter les portions de repas consommées en une journée, tout en maintenant l'unité de l'insuline donnée. De même, nous avons augmenté l'injection de l'insuline et tout en augmentant ou en maintenant la quantité d'hydrate de carbone. Cela aide à déterminer leurs effets indésirables sur l'état du patient.

Jour	Temps de repas	CHO (g)	Temps d'injection	Unité d'insuline (IU)	IU/Jour manuelle
1	8:00	45	8:00	2	8
	12:00	70	12:00	3	
	19:00	70	19:00	3	
2	8:00	45	8:00	2	8
	12:00	100	12:00	3	
	19:00	70	19:00	3	
3	8:00	45	8:00	2	10
	12:00	100	12:00	5	

	19:00	70	19:00	3	
4	8:00	75	8:00	4	16
	12:00	100	12:00	6	
	19:00	100	19:00	6	

Tableau 4: in-silico table de données

Notant que, les repas et les doses d'insuline correspondantes calculées selon un rapport insuline-glucides (ICR) 1:10, ont été administrés conformément au schéma dans le tableau 4. Figure 23 montre les données simulées, le niveau de glucose mesurée pendant 4 jours doit être maintenue entre les lignes solides et pointillées. Les 2 lignes pointillées forment la limite que la courbe doit respectée pour que le niveau de glucose soit dans le cas normal (70 - 100 mg/dL). De même pour les lignes solides (50 - 180 mg/dL), forment le minimum et le maximum que le niveau de glucose doit respecter avant d'être considéré dans le cas diabétique. Nous avons remarqué que les résultats ont la courbe dans une partie de cette simulation élevé ou faible en raison de paramètres d'entrée spécifique.

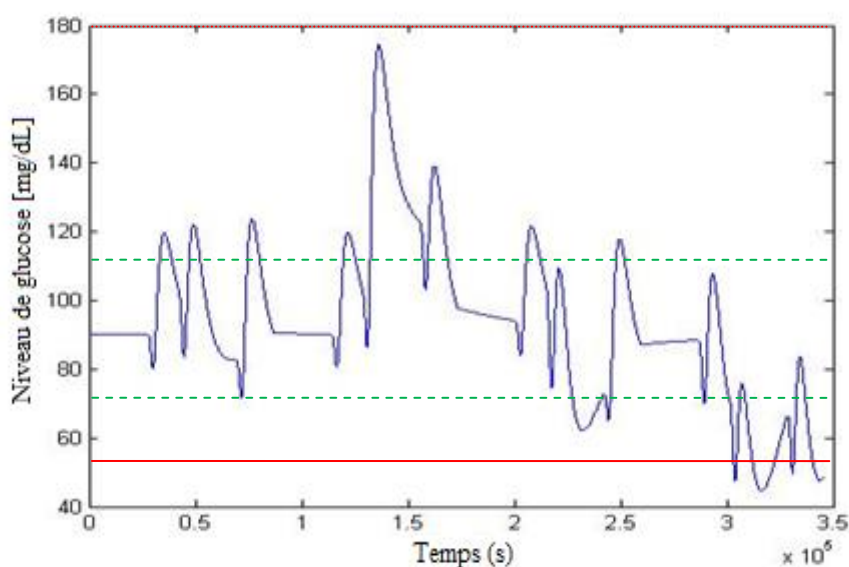


Fig. 23: Simulation des données du patient pendant 4 jours

La simulation pendant 4 jours est si importante pour notre testeur, car la simulation dans une vie quotidienne ne peut pas donner une réponse précieuse. C'est peut-être lors d'une simulation de la vie quotidienne, une partie de cette journée, par exemple pendant le petit

déjeuner, le niveau de glucose est tellement élevé et pendant le dîner est si faible. Mais au total de la QoS est en bonne valeur.

Donc, pour avoir une meilleure valeur de qualité de service, nous avons essayé d'illustrer nos résultats à partir de 4 jours ensemble et pour chaque jour au cours de cette simulation, on a comparé les résultats pour avoir plus de précision et plus d'efficacité.

3.4. Génération des patients virtuels

Le modèle utilisé Hovorka a été choisi en fonction des besoins de la recherche. Comme évoqué précédemment, il s'agit d'un modèle complet du système glucose-insuline lors d'un repas, qui a été développé en utilisant des traceurs de glucose.

On a choisi comme entrée huit variables et on a fournit à chacun d'eux des valeurs différentes afin de simuler le modèle Hovorka. Le $(Q_1, Q_2, S_1, S_2, I, x_1, x_2, x_3, t)$ aide à générer des données tout en résolvant le modèle mathématique Hovorka. En effet, la résolution d'équations mathématiques pour chaque paramètre est utile pour avoir des résultats "cliniquement précises".

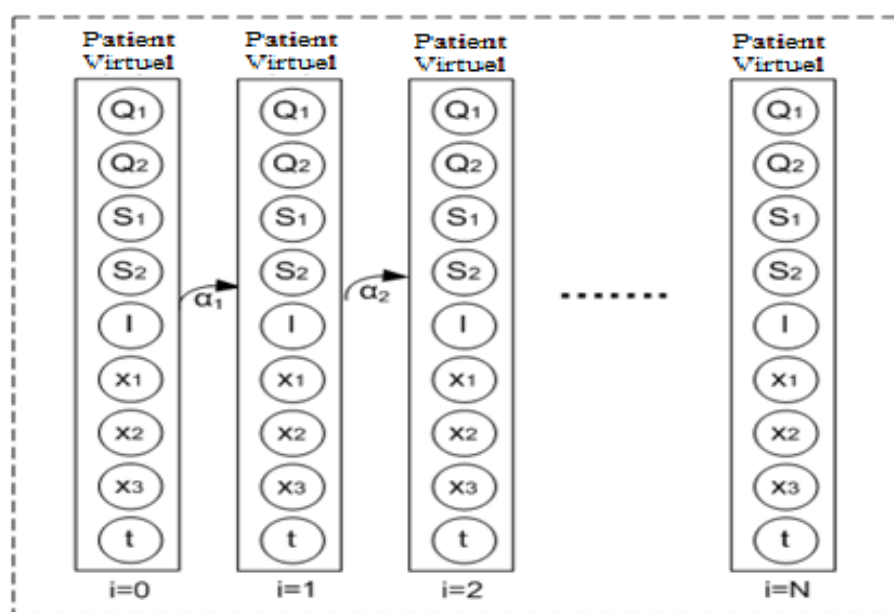


Fig. 24: Patients virtuels.

Une simple modification dans chacun des paramètres aide à avoir différents états du patient. La sévérité d'un patient peut être modifiée en changeant les valeurs des paramètres. L'objectif est d'avoir plusieurs scénarios afin de les utiliser dans le cadre de l'implémentation. Nous avons utilisé la pertinence des "processus partiellement observables de Markov décisionnels" (POMDP) [88] pour la formalisation de la planification de la gestion clinique

(figure 24), où α_i sont les valeurs de modification lors de la génération de ces patients virtuels, en tenant compte de la gamme de rapprochement de chaque paramètres. L'utilisation de la méthode POMDP décrit un procédé de contrôle stochastique avec des États partiellement observables. Nous avons commencé avec un patient virtuel et après le calcul avec des fonctions de normalisation afin de savoir dans quelle zone le point du patient est tracé, nous avons arrivé à spécifier une modification des paramètres afin d'arriver au deuxième état qui représente un autre patient virtuel. La modification peut être réalisée sur un paramètre ou plusieurs paramètres à la fois. Cette modification mineure est efficace pour générer les patients.

Le flux des processus pour générer et simuler le système de réglementation glucose-insuline est illustré à la figure 25. L'utilisation de Keil et un programme Vb.net simple aide à tester et à simuler notre système, qui peut être très utile dans des domaines des essais. Ainsi, on a utilisé pour le modèle mathématique Hovorka un vecteur d'entrée des données pour avoir beaucoup de patients virtuels et plus tard un testeur efficace. Pour nous aider à générer automatiquement ces patients, nous avons développé un programme simple qui génère approximativement les variables pour chacun des paramètres qui composent ce modèle.

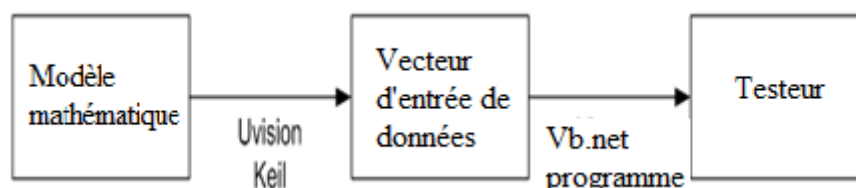


Fig. 25: Présentation des méthodes utilisées

Figure 26 ci-dessous montre l'ensemble des patients virtuels générés en utilisant notre programme simple. Notant que, chaque élément du vecteur (Q_1 , Q_2 , S_1 , S_2 , I , x_1 , x_2 , x_3 , t) peut être calculé (chapitre II modèle de Hovorka) en utilisant ensemble de paramètres. Les F_{01} , K_{12} , K_{a1} , K_{a2} , K_{a3} , K_{b1} , K_{b2} , K_{b3} , K_e sont les paramètres effectifs qui touchent l'ensemble du système.

Générer des patients virtuels

choisir un modèle **Hovorka**

F01	K12	Ka1	Ka2	Ka3	Kb1	Kb2	Kb3	ke
0.0075	0.070	0.050	0.060	0.030	2.56e-4	4.92e-5	1.56e-3	0.138
0.0075	0.070	0.100	0.060	0.030	2.56e-4	4.92e-5	1.56e-3	0.138
0.0075	0.070	0.100	0.90	0.030	2.56e-4	4.92e-5	1.56e-3	0.138
0.0075	0.070	0.85	0.25	0.030	2.56e-4	4.92e-5	1.56e-3	0.138
0.0075	0.070	0.75	0.45	0.030	2.56e-4	4.92e-5	1.56e-3	0.138
0.0079	0.070	0.068	0.052	0.030	2.56e-4	4.92e-5	1.56e-3	0.138
0.0079	0.070	0.100	0.065	0.030	2.56e-4	4.92e-5	1.56e-3	0.138
0.0079	0.070	0.100	0.84	0.030	2.56e-4	4.92e-5	1.56e-3	0.138
0.0079	0.070	0.81	0.25	0.030	2.56e-4	4.92e-5	1.56e-3	0.138
0.0079	0.070	0.67	0.45	0.030	2.56e-4	4.92e-5	1.56e-3	0.138
0.0075	0.066	0.0100	0.090	0.030	2.56e-4	8.20e-6	3.12e-3	0.143
0.0079	0.066	0.0100	0.090	0.030	2.56e-4	8.20e-6	3.12e-3	0.143
0.0075	0.066	0.0100	0.090	0.060	5.12e-5	7.38e-5	3.12e-3	0.138

Fig. 26: Programme pour générer des patients

L'environnement de simulation est constitué d'un ensemble de patients virtuels, d'un modèle mathématique et d'un modèle d'administration d'insuline ou d'un actionneur. L'algorithme de commande est en interaction avec l'environnement de simulation.

Dans la figure 27, le tracé de chaque simulation est représenté comme un point dans une grille, puis divisé cette grille en zones définies par leur x et y coordonnées, ce qui aide à construire une nouvelle grille d'erreur et l'utiliser pour évaluer l'exactitude de la valeur mesurée. Dans la première phase, l'implémentation se fait pour tester le contrôleur afin d'analyser sa performance. Notant que, le patient virtuel utilisé pour les simulations ne comprend pas les facteurs qui influent sur la concentration de glucose dans le sang, comme par exemple le stress, ou la personne faisant de l'exercice. La fonction de normalisation utilisée pour calculer le maximum BG (97.5th) et le minimum GB (2.5th) est le percentile.

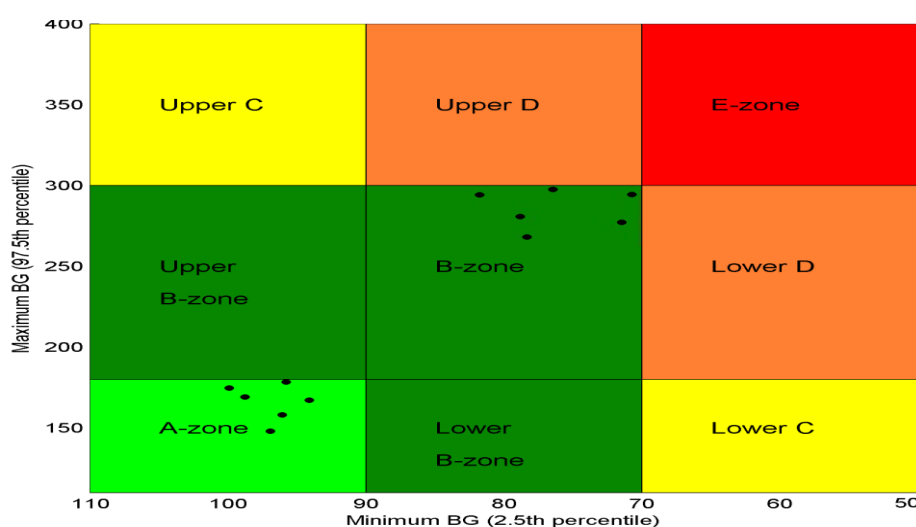


Fig. 27: Représentation graphique

On remarque, que le système d'injection automatique n'est pas considéré comme mauvais. En effet, on n'a pas un contrôleur idéal, mais nous avons arrivé à simuler la performance des équipements. L'idée principale est de considérer les cas critiques sur l'hypoglycémie et l'hyperglycémie, les capteurs, les actionneurs critiques et les contrôleurs.

Pour être plus précis et pour une meilleure représentation, chaque simulation n'est pas représentée comme un seul point dans la grille, mais elle est construite à partir d'un point central et de 3 autres points qui est l'état de la représentation du patient pendant le petit déjeuner, le déjeuner et le dîner. On a comparé ces valeurs afin de savoir que si les trois points sont dans la même zone que le point principal, cela signifie que le point principal est considéré comme la valeur désirée. En revanche, si l'un des trois points est hors de la zone du point principal, cela signifie que l'état du patient a été modifié pendant la journée et une étude plus approfondie doit être fait pour surveiller et de révéler son état.

Conclusion

Nous avons fourni une architecture flexible et globale d'un modèle de corps humain avec une représentation mathématique. Une représentation complète des composants de simulateurs, de l'analyseur, de l'indicateur schéma QoS et de la méthode d'évaluation a été réalisée. Le but de cette architecture est d'avoir un milieu complet avec la possibilité de simuler des équipements médicaux, et de tester leur performance.

Le testeur que nous avons créé est un outil efficace qui améliorera les essais dans le domaine biomédical sans imposer un risque sur les patients. Le tracé de chaque simulation comme un point dans une grille, puis la division de chaque grille en zones définies par leur x et y coordonnées, nous a aidé à construire une nouvelle grille d'erreur et à l'utiliser pour évaluer la précision des mesures de niveau de glucose faites par les patients.

Nous avons créé un modèle programmé qui agissent comme un pancréas artificiel. Le but est d'avoir un exemple complet d'un système biomédical en lien avec le glucose-insuline afin d'avoir la capacité de faire des stratégies de test. De même, nous avons défini des indicateurs simples pour les tests de niveau système. L'objectif de ces indicateurs est d'avoir une référence à la représentation graphique des données d'une personne dans le cadre du glucose.

Aussi, on a fourni un modèle qui représente une vue globale des équipements biomédicaux, basée sur des simulations répétées pour minimiser l'erreur. Les résultats ont montré la possibilité de co-simulation, et donc la possibilité de valider un système finement incorporé. Nous avons mis en place un environnement permettant la co-simulation du modèle glucose-insuline, et l'application technique in-silico en utilisant un simulateur d'un système microcontrôleur.

Dans la partie implémentation, nous avons mis en œuvre des modèles dans une approche globale afin d'associer les choix faits sur le système biomédical des indicateurs de "bien-être" associé avec les patients. Et on a simulé en utilisant une expérimentation in-silico qui présente de nombreux avantages en fournissant une plus grande productivité du travail, le coût minimum et simulation plus précise en raison de modèles sophistiqués. La simulation de patients virtuels avec le diabète de type I est effectuée pendant 4 jours, en recevant le petit déjeuner, le déjeuner et le dîner tous les jours.

Nous avons généralisé le passage des essais cliniques; en générant des patients virtuels afin d'avoir plusieurs scénarios pour l'analyse des performances. Nous avons créé un outil simple pour générer de nombreux patients en tenant compte de la gamme de rapprochement de chaque paramètre.

Un choix d'un testeur plus sophistiqué par l'introduction de différents capteurs et actionneurs apparaît comme une direction de recherche prometteuse. L'extension de l'algorithme pour des applications industrielles avec microcontrôleur ayant une capacité de calcul limitée exige un effort d'enquête non-trivial.

Table of contents

Acknowledgments	iii
Abstract	iv
Résumé	vi-1
Table of contents	vii
List of Acronyms	x
List of Tables	xii
List of Figures	xiv
Chapter I: Introduction	1
I.1. Introduction	1
I.1.1. Motivation	1
I.1.2. Objectives	4
I.1.3. Thesis structure	4
Chapter II: Systems modeling	6
II.1. Introduction	6
II.2. Architecture of the model	8
II.3. Mathematical Representation of human body	10
II.3.1. Prototype definition	10
II.4. Simulation representation of the medical system	12
II.4.1. μ Vision Keil	13
II.4.2. Simulators	15
II.5. QoS indicators	16
II.5.1. QoS schema	17
II.5.2. QoS evaluation	20
II.6. Conclusion	27

Chapter III: Case study - Systems implementation	28
III.1. Introduction	28
III.2. Glucose–Insulin System	30
III.2.1. Diabetes mellitus overview	32
III.2.2. Glucose Level	33
III.2.3. Closed-loop	34
III.3. Models Implementation	41
III.3.1. Implementation of Human body	41
III.3.2. Implementation of artificial pancreas	47
III.3.3. Implementation of sensors and actuators	51
III.4. Programmed Software	54
III.5. Models and methods of experimentation	55
III.5.1. Experimental conditions	56
III.6. Conclusion	59
Chapter IV: Simulations results & QoS measurement practices	60
IV.1. Introduction	60
IV.2. Clinical cases generation	61
IV.2. Virtual patients sample	63
IV.3. Simulation Implementations	64
IV.3.1. Virtual patient one	67
IV.3.2. Virtual patient two	68
IV.4. QoS evaluation implementation	69
IV.4.1. Graphic representation	70
IV.4.2. Grid representation	71
IV.5. Conclusion	74
Chapter V: Conclusion	75

V.1. Contribution.....	75
V.2. Future work	76
V.3. Publications	77
APPENDIX A	88
APPENDIX B	91
APPENDIX C	96

List of Acronyms

The following acronyms and abbreviations are used in this thesis.

AP	Artificial Pancreas
ADA	American Diabetes Association
APM	Artificial Pancreas Microcontroller
BG	Blood Glucose
CGM	Continuous Glucose Monitor
CGMS	Continuous Glucose Monitoring System
CHO	Carbohydrates
CG-EGA	Continuous Glucose–Error Grid Analysis
DM	Diabetes Mellitus
DS	Device simulation
EGP	Endogenous Glucose Production
EGA	Error Grid Analysis
FDA	Food and Drug Administration
GOx	Glucose oxidase
GDH	Glucose dehydrogenase
GWB	GlucoWatch® G2TM Biographer
IDDM	Insulin-dependent diabetes mellitus
IU	Insulin Unit
IDF	International Diabetes Federation
ICR	Insulin-to-carbohydrate ratio
ISO	International Organization for Standardization

IVGTT	Intravenous Glucose Tolerance Test
NIDDM	Non-insulin-dependent diabetes mellitus
POMDP	Partially Observable Markov Decision Processes
PS	Patient simulation
QoS	Quality of service
SMA	Shape Memory Alloy
Std	Standard deviation
Stderr	Standard error
T1DM	Type 1 diabetes mellitus
WHO	World Health Organization

List of Tables

Table II. 1: Error grid zones definition.	25
Table III. 1: Interpretation and approximate value of BG level.	34
Table III. 2: Hovorka model parameters.	39
Table III. 3: Hovorka Function definition.	42
Table III. 4: Hovorka function implementation algorithm (1).	43
Table III. 5: TestGB function definition.	44
Table III. 6: Hovorka function implementation algorithm (2)	44
Table III. 7: Testins function definition.	45
Table III. 8: Hovorka scenario implementation algorithm.	47
Table III. 9: Control algorithm main rules.	49
Table III. 10: Specific parameters.	49
Table III. 11: Glucose test before and after meal.	53
Table III. 12: Function that Calculate the computed dose.	54
Table III. 13: In-silico table data.	57
Table IV. 1: 6 virtual patients' data.	64
Table IV. 2: Percentile function definition.	64
Table IV. 3: Percentile Algorithm.	66
Table IV. 4: Median function definition.	66
Table IV. 5: Median algorithm.	66
Table IV. 6: Standard deviation function definition.	67

Table IV. 7: Standard deviation function implementation.	67
Table IV. 8: $IG_H(g)$ with α, β predefined constant values.	71
Table IV. 9: Clarke4 function definition.	72
Table IV. 10: Grid function algorithm.	72

List of Figures

Fig. II. 1: Interdependence of body systems.	9
Fig. II. 2: Physiological regulation.....	9
Fig. II. 3: Core simulator components.....	13
Fig. II. 4: Simulator.....	15
Fig. II. 5: Analyzer	16
Fig. II. 6: QoS indicator schema.	18
Fig. II. 7: QoS input and output.	18
Fig. II. 8: Closed loop system model.	19
Fig. II. 9: Tester implementation using closed loop.....	20
Fig. II. 10: Closed loop system overview.....	21
Fig. II. 11: Index of severity representation	21
Fig. II. 12: Clarke and Parkes error grid for glucose.....	23
Fig. II. 13: Error grid graphic representation.	24
Fig. II. 14: Graphic representation using normalization function.	25
Fig. II. 15: Simulate the performance of equipments.....	26
Fig. III. 1: The blood glucose–insulin system.....	30
Fig. III. 2: Interaction schema of the integrated system.....	31
Fig. III. 3: Blood glucose level presentation.	34
Fig. III. 4: Closed Loop system graph.....	35
Fig. III. 5: Minimal glucose model.	37

Fig. III. 6: Hovorka model.	38
Fig. III. 7: Man-Rizza-Cobelli model.....	40
Fig. III. 8: Insulin given before the meal.....	47
Fig. III. 9: Simulation closed loop.....	48
Fig. III. 10: With and without insulin injection.....	50
Fig. III. 11: Simulations results: (a) +1 unit injected (b) +2 units injected.....	51
Fig. III. 12: Configuration of a biosensor.....	52
Fig. III. 13: Micro-pump & micro-valve with piezoelectric actuator.	52
Fig. III. 14: Vb.net program.	55
Fig. III. 15: Global view of biomedical equipments.	55
Fig. III. 16: In-silico with Hovorka model.	56
Fig. III. 17: Simulated patient data during 4 days.....	58
Fig. IV. 1: Virtual patients.....	62
Fig. IV. 2: Outline of the methods used.	62
Fig. IV. 3: Software program to generate patients.	63
Fig. IV. 4: Software program to calculate percentile, mean, std, stderr, median.	65
Fig. IV. 5: Virtual patient one.	68
Fig. IV. 6: Normal Virtual patient (1).	68
Fig. IV. 7: Virtual patient two.....	69
Fig. IV. 8: Normal Virtual patient (2).	69
Fig. IV. 9: Graphic representation.....	73

Chapter I: Introduction

I.1. Introduction

This introductory chapter presents an overview of the thesis, beginning with the general considerations that motivated this research. The main objectives, and the methodology used in this study are briefly explained. Finally, a description of the structure and content of the thesis is presented.

I.1.1. Motivation

There is a long history of biology systems and in the concept of system theory and classical physiology [1, 2, 3]. The current biological systems consist of large numbers of components. Studying the components parts and also focusing on understanding its structure, help to understand the function and behavior of such systems [4]. It is also known that these proprieties and the function of these systems emerge through the interactions of the components [5, 6]. On each level of this system, this behavior depends on the proprieties of these levels, the outputs and their interaction. For this reason, we need a method for simultaneously studying the different levels of the system. An important key part of the systems biology approach is the computational and Mathematical modeling, which help to produce models for describing systems [7].

Modeling is fundamental and there are many different ways to understand, define, and analyze the structure of the complex system and phenomena. Advances in information technologies and systems, reflect an increasing in the domain of medical devices, which leads to better diagnosis and delivery of treatment, enhancement in usability and new functionalities. The importance of simulation modeling in medical instrumentation is arriving to reduce medical errors, solve health problem and to improve patient safety. The issue of operational test and safety of these devices remains problematic and difficult to solve. Several parameters (patient, disease progression, treatment, etc.) are indeed taken into account and the impact on the patient can be catastrophic.

Recent models in the literature suggest that the simulation modeling techniques are useful tools for analyzing complex systems in critical care [8, 9, 10]. For example, modeling the

glucose-insulin interaction, in case of diabetes mellitus (DM), requires an understanding of the physiological and metabolic processes that determine the observable behavior [11, 12, 13]. The DM is described as a metabolic disease which is characterized by high or low plasma glucose level which is a major cause of death in most countries. According to the International Diabetes Federation (IDF), for the years between 2010 and 2030, close to four million deaths group can be attributed to diabetes. There is a need to reduce, or at least contain, the health care costs.

Previously, without simulation, there was a delay in work progress and the cost was more expensive. There was a need for providing higher work productivity and minimum cost. Simulation modeling enables to virtually investigate many prototypes and analyze all inputs and outputs, constraints and device behaviors. The simulation was defined as a technique or a method, used in health care education fields [14] and assessment, to replace real patient with scenarios designed to promote knowledge and experiences. Noting that, the goal of mathematical simulation is to develop an understanding of integrated physiological systems.

Biomedical simulation test [15, 16, 17] is a new type of medical test which is a kind of simulation medical procedure performed to detect, diagnose, analyze, or monitor biomedical equipment [18, 19]. The test is used to assess scientific aptitude of biomedical equipment. The performance of simulation must be evaluated to determine whether it is optimized or not. Hence, in measuring the quantitative performance of the simulation, mean error is implied which was to determine the error between the targets and the output. Innovations in biomedical technologies are seen as being able to provide solutions to improve the quality and the efficiency of healthcare systems [20].

There is a need to improve the quality of the services (QoS) provided, by ensuring biomedical devices are fit for purpose, which give an opportunity to develop new services or new diagnosis with an objective of upgrading and improvement. In other words, a set of quality of services must be satisfied. The problem that should be solved is how to analyze embedded medical devices in order to avoid risks on patients, including variability and uncertainty on a number of parameters related to patients, evolution of disease, and treatments. Treatment depends on the availability and well functioning of complex electronic systems, including thousands lines of codes.

Today medical treatment uses more and more embedded devices including sensors [21, 22, 23], actuators [24, 25] and controllers [26]. Treatment depends on the availability and the well functioning of complex electronic systems, including thousands of lines of codes.

Products of all kinds are increasingly constructed to include one or more embedded systems to improve functionality. These embedded systems often combine hardware and software that together allow a distinction. Depending on the scope, embedded systems present a wide variety of formats and technical solutions. Embedded systems, distributed real-time and manipulate streams or media critics, spread more and affect all sectors (health, recreation, avionics, telecommunications, transport ...) [27]. These systems have strong requirements in terms of security, fault tolerance and quality of service. Issues related to QoS are not simply those of improving performance. These are mostly managing the resources of the support system to meet different QoS requirements. Testing embedded system consists of evaluating the application behavior (including its components), performance, and robustness [28, 29].

One of the reasons for the high growing in research simulations fields is the ability to have data. Various experiments grow due to faster and better methods used to obtain this amount of physiological data. This data constitute a bank of potential insight that provides us with more statistical analysis that helps to discover correlations.

First, there is a need to propose a mathematical model for body representation and a simulation tool for health management, which help to provide a new comprehensive toolset to tackle the issues of system modeling, analysis, QoS system integration and verification. Also a mathematical model is needed to represent physiological system, which describes how a system works, how organs function, and a method to deal with medical data collected.

Second, in medical fields, it's important to improve the diagnosis and treatment of patients, and analyze the performance of embedded biomedical devices; by showing its validation which helps to make safety its operation.

Third, there is a need to calculate the quality of services of such system, as well as to find a way to prove that a sensor or an actuator is more effective than another.

The method and tools developed here can be used by patients and physicians to evaluate the usage of biomedical devices and to have a complete environment for simulation and testing.

I.1.2. Objectives

The general objective of this research is to develop tools and methodologies for the development of embedded systems for medical fields. The goal is to be able to model and jointly simulate the medical device as well the human body, at least the part of the body involved in the medical device, to analyze the performance and quality of service (QoS) of the interaction of the device with the human body.

To achieve this objective, the study addresses the following specific aims:

- To propose a new prototype that helps to represent the human body model in order to better understand all the systems that the human body is composed of. In addition, understanding and simulating them in order to detect performance and quality of service of all system components.
- To develop a complete environment of biomedical system in order to have the ability to test testing strategies, a powerful implementation that helps to test these biomedical devices, and a bank of test can then be created. The ability to develop a model in order to analyze the performance of embedded biomedical devices can be very useful in medical research which leads to improve the diagnosis and treatment of patients. We can generate many models and disturbed parameters to reproduce many states that might resemble to physiological cases in disease processes.
- To construct a complete realized hardware environment that has the ability to improve the modeling and analysis approaches of embedded systems in order to make them faster and less expensive.

I.1.3. Thesis structure

This thesis is divided into five chapters. Here, a detailed overview of all the chapters of the dissertation is described:

- **Chapter I** outlines the motivation for this research, the problems and challenges involved, and the research objectives.
- **Chapter II** starts by defining a prototype of a new global and flexible architecture of mathematical model of human body that is able to contain required data. Next, we

propose a new global methodology for modeling and simulation human body and medical systems, in order to understand thoroughly the best way to model and simulate these systems as well as detecting performance and the quality of services of all system components. Finally, the techniques used to help evaluate the calculated QoS value.

- **Chapter III** presents the implementation part of a case study where we use a mathematical model applicable in our prototype that was described in chapter II. Then, we take the glucose insulin system and create a new framework in order to test the performance of all system components. As well as, the simulation set-up, the explanation of the modeling work and the presentation of the control algorithm. Thus, we have presented the three models implementations.
- **Chapter IV** is for the simulations results and the QoS measurement practices. It shows virtual patients generating part, modeling results as well as the control performances in closed-loop achieved exploiting the in-silico patient. This chapter also shows the performance of our tester model declared in chapter II for the case studied in chapter III which is the glucose level.
- **Chapter V** discusses the conclusions and contributions of this research, and future work.

Chapter II: Systems modeling

II.1. Introduction

With the rapid development in medical research, the study of simulating human body is moving towards more and more precise methods. Since the beginning, the scientists have been interested in the physiological system of the human body and its applications. They have tried to find the most accurate model in order to simulate the human body and all the reactions with his environment. The world health organization defines medical devices as “any instrument, apparatus, implement, machine, appliance, implant, in vitro reagent or calibrator, software, material or other similar or related article, intended by the manufacturer to be used, alone or in combination, for human beings for one or more of the specific purposes of: diagnosis, prevention, monitoring, treatment or alleviation of disease; investigation, replacement, modification, or support of the anatomy or of a physiological process; control of conception” [30].

The study of the structure of the human body and its parts is called anatomy while physiology is the study function of these parts. Anatomy is a broad field of study that includes many specialties; each of them would be a complete domain research. These specialized divisions of the anatomy are especially useful for scientific research and diagnosis of diseases. Like anatomy, physiology includes several specialties, but physiology highlights the dynamic nature of the organization whereas anatomy gives a static image of the body.

These two complementary scientific disciplines, affect the fundamentals that allow us to understand the human body. The study of the function and the structure are inseparable. Indeed, the function always reflects structure. That is to say that an organ can only perform the functions permitted by its structure, noting that an organ is a structure composed of at least two types of tissues that exerts a specific function in the body. Tissues are groups of cells that perform the same function. For example, the heart can pump blood only in one direction because of the structure and arrangement of the heart valves.

Each organ has a specialized functional structure that performs an essential activity that no other body can perform instead. These construct level of systems; each system consists of

organs that work together to perform a single function. All systems cannot work completely independently; they all work as one organism (for example, digestive, cardiovascular, muscular and respiratory systems). Noting that, all systems work together to maintain human life, by maintaining controlled factor or what we can call physiological variables. These variables must be maintained by the system that analyzes the data it receives and then determines the appropriate response.

We can resume the variation of these variables as follow; an external event affects a physiological variable, a receptor sensor detects this modification, and sends an input signal about this detection to the system to control it. The control system sends an output signal to the effectors' actuator, the signal sent by the actuator acts on the intensity of the stimulus feedback. Simulation this procedure leads to a better understanding of the whole system.

In this chapter, we propose a global methodology for modeling and simulation medical systems and human body, in order to analyze the performance and the quality of services of all system components. We begin first by defining a new prototype of a global and flexible architecture of mathematical model of human body which is able to contain the required data. We describe next the simulations representation, by mentioning in details the core simulator components, analyzer, and the quality of services indicators. The simulation of the mathematical models provides useful tools for the diagnosis and analysis the interactions between efficacy, therapies, side-effects, and outcomes. This will help to better understand the human organism control, to analyze experimental data, to identify and quantify relevant biophysical parameters, and to design clinical trials.

II.2. Architecture of the model

Many research centers are developing environments for developing integrative models of human physiology. Biomedical researchers can use integrative physiological models to better understand the fundamental relationships hidden in the complexity. Computational methods have been used to study and describe physiological responses [31], there are few examples of integrative models of human physiology. They studied model aspects of the heart, cardiovascular, and pulmonary systems [32].

Later, they provide an analysis of the starting point for the development of multi-level systemic biological analysis, combining reduction and integration [33]. Humans must keep their limits, move, react to changes in their environment, ingest and digest food, have a metabolic activity, eliminate waste, reproduce and grow. The distribution of vital functions, between different systems, leads to interdependence of all body cells [34, 35]. The human body has several levels of complexity. For example using the cardiovascular system, different levels of integration complexity of the human body are illustrated. Every system appears to be the result of a combination of several levels of integration.

Figure II.1 shows the interdependence of body systems, where the integumentary system protects the whole body against the environment. There exists a membrane which forms an envelope permitting the useful substances but preventing the passage of unnecessary or harmful substances. The cardiovascular system is responsible for transporting nutrients and removing gaseous waste from the body.

The digestive system and respiratory system interact with the environment and provide respectively nutrients and oxygen to the blood which then distributes to all cells. Metabolic wastes are eliminated from the body through the urinary system and the respiratory system. None of the systems works completely independently, they all work for the well-being of the whole organism.

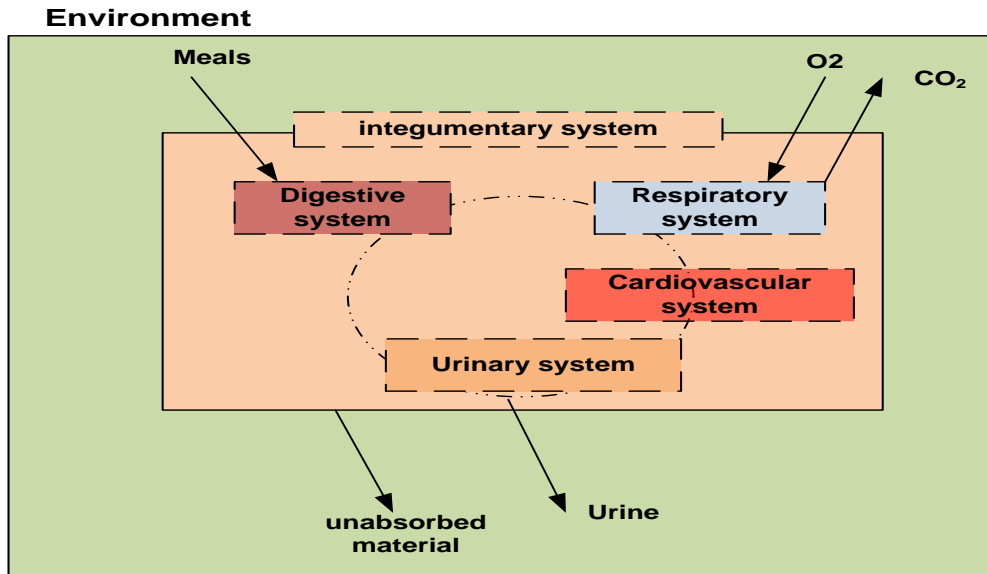


Fig. II. 1: Interdependence of body systems.

In the human body there are a lot of physiological variables that must be maintained or controlled. Let *PV* be the abbreviation concerning physiological variable. Such as: blood pressure, acidity of the blood, blood sugar level, heart rate, body temperature, and breathing rate. Any event that affects the system reflects a change in the physiological variable, and the system tends to maintain conditions that require frequent monitoring and adjustment within physiological limits.

The blood pressure is a *PV* controlled by the human body. When an event causes this *PV* to increase, pressure-sensitive nerve cells (sensors), in certain arteries, send nerve impulses (input) to brain (controller). The brain interprets the messages and responds by sending fewer nerve impulses (output) to the arterioles. This causes the arterioles (actuator) to dilate (response) (figure II.2).

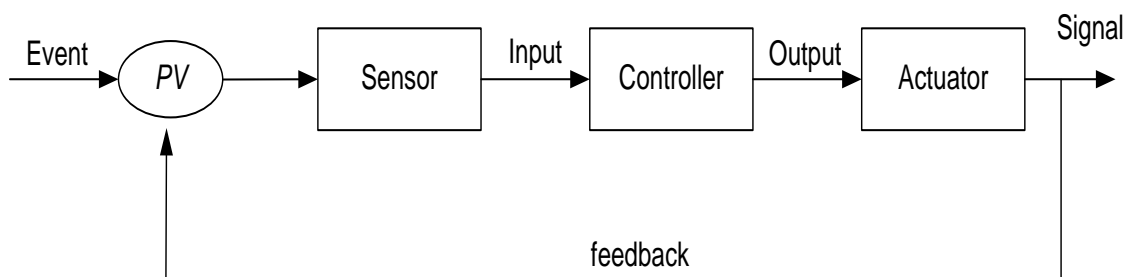


Fig. II. 2: Physiological regulation

All *PV* needs relatively to be in stable conditions in order to live and consequently the human systems perform their functions to keep alive the whole body. Figure II.2 above shows how the human body can be interconnected to form a physiological closed-loop system. It is basically a sensor whose role is to monitor the environment and respond to changes of the *PV*, sending information (input) to the controller.

The controller, which sets the reference value where the variable must be maintained, analyzes the data it receives and determines the appropriate response. Then, the Actuator through which the controller produces a response (output) to the event. The response then produces a feedback action which acts on the event; it may have either a reducing or strengthening effect so that all the regulatory mechanism either ceases to operate or the reaction proceeds with increasing intensity.

Sometimes the controlling is done by changing the value of *PV* in the opposite direction of the initial change and returns it to a normal value.

II.3. Mathematical Representation of human body

The use of ordinary, partial, and integral differential equations to model biological systems has a long history. Mathematical modeling is becoming an increasingly important subject that helps to expand our ability to translate mathematical equations and formulations into concrete conclusions [36, 37]. Species distribution models have been used extensively in many fields [38]. The mathematical model of a patient is a set of equations and various algorithms, where the use of parameters and constants to resituate the human body functioning. We define a new prototype to represent the mathematical model of the human body, described in details in the section below.

II.3.1. Prototype definition

Let Mp be a mathematical model for a patient, S is a set of system, Pf a set of personal information about the patient that may affect the system such as weight, stress, etc..., Dm a set of daily meals, Or a set of body organs, Ir a set of internal relations for the body actors, Er a set of external relations between the body actors, C a collection of constants related to systems actors, and Par a collection of parameters. PV is a collection of physiological variables reflecting the state of a patient. Both C and PV are numerical values that may be used by the relations (equations) in the model. SC is a set of constants and SPV is a set of physiological variables where SC is a subset of C and SPV a subset of PV .

Let W is the weight, St is the symbol to indicate stress, Sp to indicate if the person do sport, etc... The set Pf is defined by:

$$Pf = \{W, St, Sp \dots\} \quad (1)$$

Let B is the set of breakfast food values such as calories fat protein etc., L is the set of lunch food value, D is the set of dinner food values, and Af is the set of additional portions values such as snacks. The set Dm is defined by:

$$Dm = \{B, L, D, Af\} \quad (2)$$

Let Or_i are an organ in human body. The set Or is defined as:

$$Or = \{Or_1, Or_2, Or_3 \dots Or_n\} \quad (3)$$

Let Ir_i are an internal relation (mathematical function) for an Or_i that uses values from $SC \in Or_i$ and $SPar \in Or_i$. The set Ir is defined as:

$$Ir = \{Ir_1, Ir_2, Ir_3 \dots Ir_n\} \quad (4)$$

Let Er_i are an external relation (mathematical function) between two or more Or_i that uses values from C and Par . The set Er is defined as:

$$Er = \{Er_1, Er_2, Er_3 \dots Er_n\} \quad (5)$$

Let $SName$ is the name of the body system (such as Digestive system), SC and SPV are related to concerned S_i . Noting that, intersections between SPV of different systems are not necessarily empty, because some physiological variables may affect several systems so they are common variables between these systems. The system S consists of one or more Or . The set S is defined by:

$$S = \{S_1, S_2, S_3 \dots S_n\} / S_i (SName, SC, SPV, Ir, Er) = \bigcup_{j=1}^n Or_{ij} \quad (6)$$

Let Par_i are input parameters for the system, $Par = U_{SPar}$, $SPar \in S_i$. The set Par is defined as:

$$Par = \{Par_1, Par_2, Par_3 \dots Par_n\} \quad (7)$$

Let C_i are predefined constants for the system / $C = U_{SC}$, $SC \in S_i$. The set C is defined as:

$$C = \{C_1, C_2, C_3 \dots C_n\} \quad (8)$$

Let PV_i are physiological variables controlled by the system / $PV = U_{SPV}$, $SPV \in S_i$. The set PV is defined as:

$$PV = \{PV_1, PV_2, PV_3 \dots PV_n\} \quad (9)$$

Finally $Mp = (Pf, Dm, S)$ would be a global and flexible prototype model applicable to any analyzer for a patient and may integrate one or many body systems constituting a case of study or even all body systems.

This model can feed (input) the device model with its needed values via the results of the relations sets (Ir and Er) as well as the physiological variables set PV .

II.4. Simulation representation of the medical system

This part is described as “Core simulator components”, shown in figure II.3, contains two simulations sections related with each other's by input/output hardware interface components which facilitate the communication between them. It presents the peripheral used to provide data and controlling signal between the two sections. The first section described as “Patient simulation” (PS) simulates the patient using a mathematical model; the patient model has been simulated to better understand the mechanisms of the human organic system. There are many constants and parameters involved in the model. There are usually decided upon collecting data or experimenting. These models are the best way to simulate patient physiology and pharmacology and provide responses in real time to whatever treatment has been given.

The second section named as “Device simulation” (DS) used to describe the simulation of medical devices using mathematical model equations, with the ability to simulate also sensor, actuator and controller. This mathematical model is used to mimic the function of medical device by simulating hardware and application, which helps to form a complete system simulation. The idea from simulation is to improve the design and testing of medical devices, which can be simple devices to be used in educational fields or complex devices that combine mechanical models with computer stations.

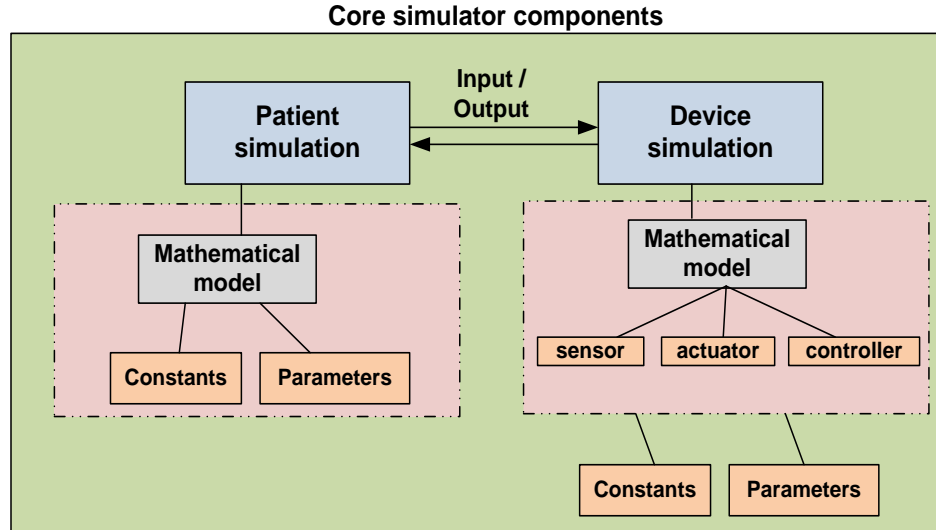


Fig. II. 3: Core simulator components.

The mathematical models have been successfully developed for testing, simulation optimization, control, design and diagnostic. The core simulator components can be represented as a tuple $\langle Mp, Md, S, A, Cr \rangle$ where Md is the mathematical model of device, S for sensor, A for actuator, and Cr for controller.

We can simulate the biomedical equipment, by simulating the patient's level, sensors and actuators levels, and simulating the electronic hardware. We have successfully implemented and simulated a mathematical model of the human body (ref. chapter III), using Keil [39, 40] development tools designed for ARM processors. Sensors and actuators were also simulated using codes that simulate their functions in chapter III.

II.4.1. μ Vision Keil

There is a need for a development tool of the entire system that must be reliable and cross-platform to support a diverse set of developers, for this reason, we have implemented using Keil μ Vision 3 development tools designed for ARM an processor-based microcontroller device; that works with embedded C language. Keil is used as software development tools for embedded microcontroller applications. It has a simulator part that simulates most features of a microcontroller without the need for target hardware. By using it, we can test, debug and simulate codes and a wide variety of peripherals. It has a powerful compiler and tools used to write the C-code for the programmable core. In this software editor, we are writing the program in Embedded C that helps to generate embedded applications for virtually every 8051 derivative. In brief, it compiles C code, assembles assembly source files, link and locate object modules and libraries, creates HEX files, and debugs the target program. Vision is an

integrated development environment that combines project management, source code editing, and program debugging in one single, powerful environment. The main advantage of using Keil in our work is to define an optimal model that works with embedded C language and can be implemented in a microcontroller. And because it allows us to debug programs using only my PC and device simulation drivers provided by Keil and various third-party developers. it simulates our entire target system including interrupts, startup code, on-chip peripherals, external signals, and I/O.

µVision3 helps to expedite the development process of embedded applications by providing the following:

- Full-featured source code editor.
- Device database for configuring the development tool setting.
- Project manager for creating and maintaining your projects.
- Integrated make facility for assembling, compiling, and linking your embedded applications.
- Dialogs for all development tool settings.
- True integrated source-level Debugger with high-speed CPU and peripheral simulator.
- Advanced GDI interface for software debugging in the target hardware and for connection to Keil ULINK.
- Flash programming utility for downloading the application program into Flash ROM.
- Links to development tool manuals, device datasheets & user's guides.

The µVision3 IDE offers numerous features and advantages that help you to quickly and successfully develop embedded applications. Noting that the screen provides a menu bar for command entry, a tool bar where you can select command buttons, and windows for source files, dialog boxes, and information displays. µVision 3 can simultaneously open and view multiple source files. This version has two operating modes:

- **Build Mode:** It allows to translate all the application files and to generate executable programs. The features of the Build Mode are described under Creating Applications.
- **Debug Mode:** It provides a powerful debugger for testing your application. The Debug Mode is described in Testing Programs.

II.4.2. Simulators

The medical definition of a simulator is “a device that enables the operator to reproduce or represent under test conditions phenomena likely to occur in actual performance” [41]. Lack of communication and lapses human error are the most potential accidents in medicine, which leads medical simulation as a powerful technique to bridge this gap.

When talking about simulator we also talk about scenarios, patient and devices (figure II.4). The body of the simulator is composed from them. In case of patient, the simulator use a more sophisticated technique using mathematical model differential equations of a patient's physiology, with specific constants and some parameters versus time for example meals. Also in case of device, the simulator use also advanced model, constants and parameters versus time if needed.

We can, for example, simulate the misuse of a particular device by either changing initial configuration or by creating a problem in utilization, will give a corruption in result. In another situation, the use of practical medical scenarios helps to illustrate effective equipment and diagnosis procedures.

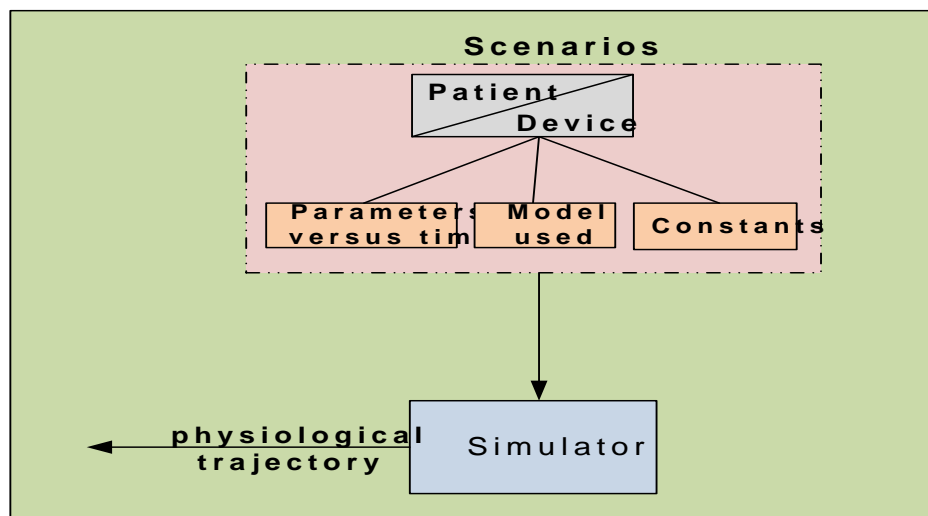


Fig. II. 4: Simulator

The simulation helps to optimizing the performance of devices avoid risks on patients and an evolution the treatment of disease. Taking many mission-critical simulations, by creating many population scenarios helps minimizing deficiency of accuracy and having a good performance and quality of service.

One of the most important uses of simulation output analysis regards the comparison of competing systems or alternative system configurations [42]. An important feature of simulation is its ability to allow the experimenter to analyze and compare scenarios quickly

and efficiently.

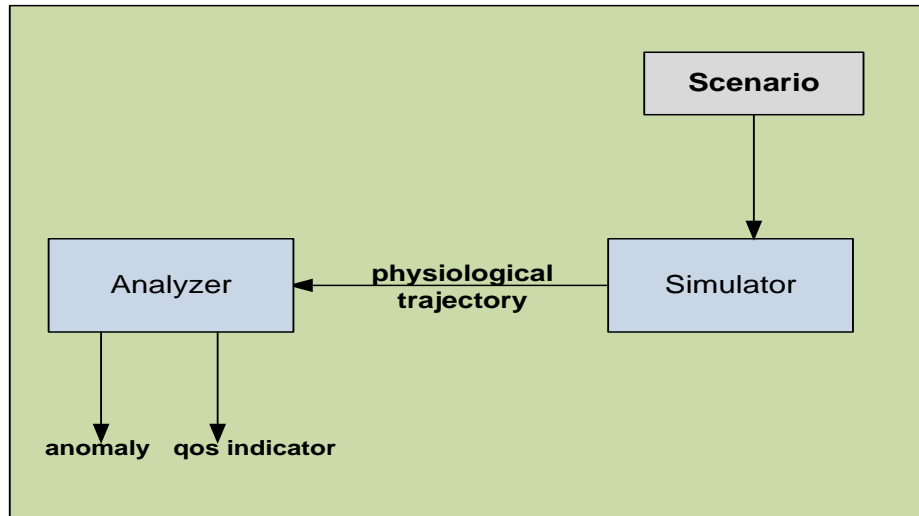


Fig. II. 5: Analyzer

Figure II.5 above describes that using simulator from a scenario guides us to generate the physiological trajectory. The analysis of this trajectory helps us to know if there is anomaly or not, and to generate values concerning the indicator of the quality of service. The objective is to provide the expected services in a dependable way, and maintaining the required Quality of Service levels. In other words, a set of Quality of Services must be satisfied.

II.5. QoS indicators

Recently, the quality of Service has received considerable scientific attention. While QoS has an important role in any system QoS has focused mainly on availability, reliability, security and cost. Most of the researches in medical applications focus on the concept of QoS and QoS requirements, measurement and management [43, 44, 45, 46]. QoS issues related to data access and retrieval is rather less investigated. Maintaining QoS is equal to maintain the *PV* in a normal state. Each *PV* controlled have a maximum and minimum levels that this value must reach without arriving to the dangerous state. Estimates calculation of the range of controlled values depends on the controlled *PV* and on some characteristics of the human body. However, this is just an estimation; individual values vary considerably from this average value. Noting that, a normal value is the reliable value that remains constant from day to day and changes only slightly from year to year.

We are alive because important *PV* in our body are regulated automatically and so remained within certain levels. We can imagine what may happen without these controls by thinking about the consequences of a control failure. The human body has its main automatic

control mechanisms. Feedback is a key concept. The actual values of PV are sensed, feedback and used to control the system. The behavior of these controlled variables is typically described by differential equations in the time domain.

When a change in a state of a PV is occurred upon an event, negative feedback responses are triggered to bring the PV back to its normal point. The sensor, controller, and the actuator play a role as a parts of negative feedback response.

The control specifications of the PV may refer to a static value or to a dynamic value that change by time. The study of each system helps to identify these variables, which need to be manipulated and by how much, in order to achieve given desired specifications.

Consider α_i range of value that PV can reach from normal value to the maximum / α_1 is the highest value. Consider β_i the range of value that PV can reach from the normal value to the minimum / β_1 is the lower value, then we can write the following:

$$\{\beta_1, \beta_2, \beta_3, \dots, \beta_n\} < PV < \{\alpha_n, \dots, \alpha_3, \alpha_2, \alpha_1\} \quad (10)$$

II.5.1. QoS schema

For each scenario a quality service is recovered. After creating and simulating millions of scenarios, we can use the information of each scenario to retrieve a more comprehensive quality of service that will not depend on the scenarios.

In fact we don't generate scenarios (figure II.6), but we create a system that allows generating multiple scenarios to retrieve more relevant information. We can modify the number of device and their characteristics (N_d), the number of patients and their characteristics (N_p), or a mix of patients, devices and parameters ($N_{s,p,d}$) of different scenarios.

Creating multiple scenarios, in order to simulate and analyze data of results. To create them, we need many population and prevalence data, and with a number of generic pathways physiological trajectory to calculate the proper quality of service (QoS).

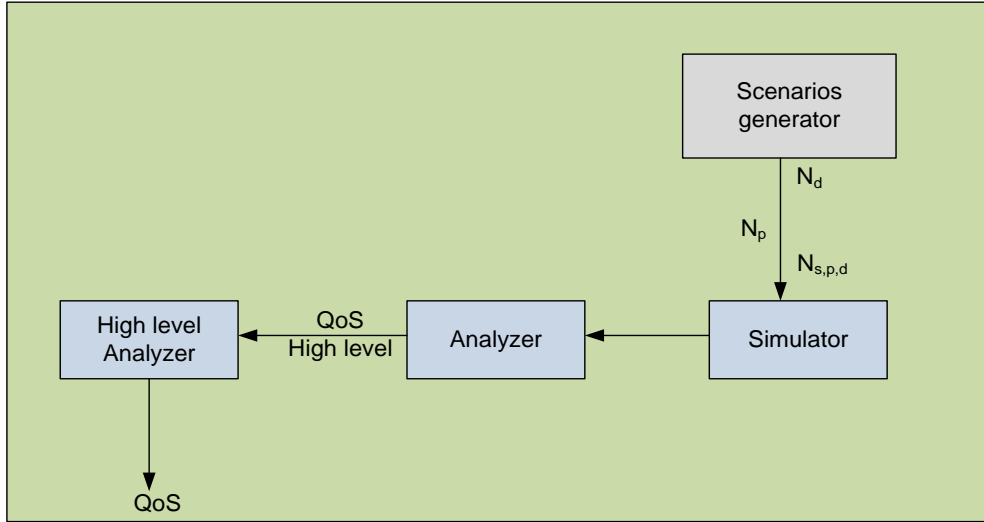


Fig. II. 6: QoS indicator schema.

Create virtual population of many cases accompanied by sets of n parameter representing n virtual cases. A population is considered as an array of cases. Each case has characteristics represented by fixed and variables parameters. For example this vector is composed like this (X_1, X_2, \dots, t) , that help to generate data while solving a mathematical model. Indeed, solving mathematical equations inside each parameter helps to have “clinically accurate” results (see Figure II.7).

An implementation of this vector is fully described in next chapter. A virtual subject with type 1 diabetes is represented by a model of glucose regulation and its parameters. The population considered as inputs, for actuators, sensors and controller, which help to calculate the value of Y considered as an approximately output value that represents QoS value of the system. In brief, for any system that has inputs parameters and output values, we can calculate a QoS value based on a population and simulated devices.

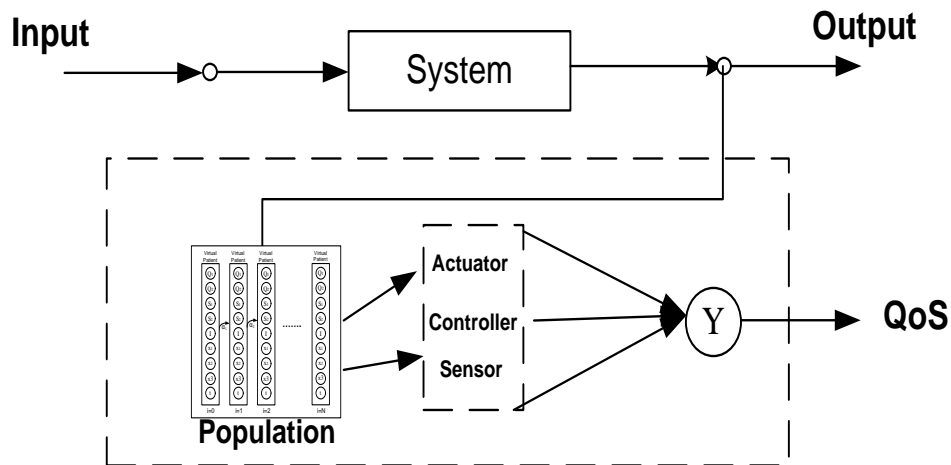


Fig. II. 7: QoS input and output.

The closed-loop system model chosen is based on feedback queue system, which allows describing the main characteristics of the model:

- The feedback system in order to improve performance analysis.
- The system takes from the input queue, as virtual patients using generation method.

A work-conserving system means that if one flow is out of packets, the next data flow will take its place. In figure II.8, the patient generation is modeled as non-FIFO because the order of the outputs results differs from the orders of inputs, and considered as work-conserving system because of the presence of (t, Par, C) and feedback. Noting that, Non-FIFO method is used in many domain fields [47, 48].

The meaning of the parameters mentioned in the figure II.8 is the following:

- S to represent the system to be controlled.
- PV the variable that the system tries to control.
- Feedback helps to monitor and report values.
- Tester is our tester model.
- (t, Par, C) composed from parameters, constants and t as time variable.
- Par represents the parameters given.
- Dm represents the daily meals taken by the patient.

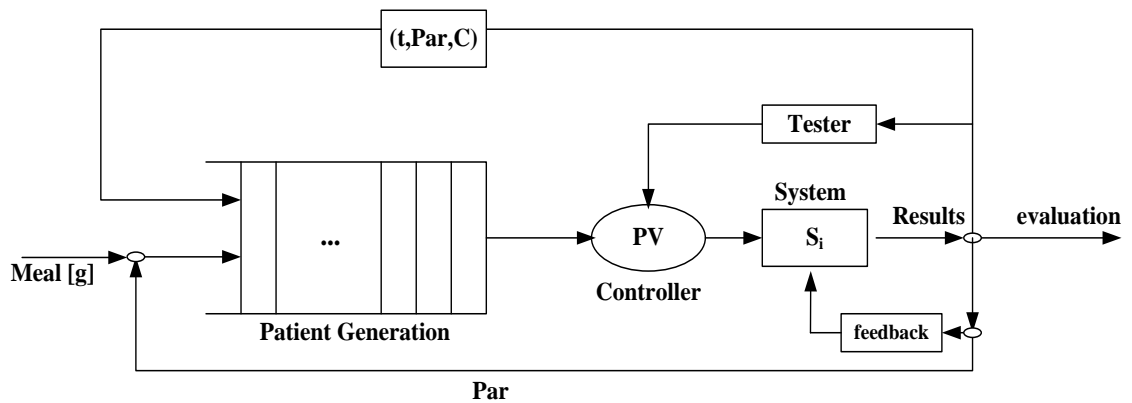


Fig. II. 8: Closed loop system model.

In a system there is a PV to be controlled and a tester that are interconnected to form a physiological closed-loop system. A high-level overview of the system allows creating many clinical cases that can benefit from closed-loop systems.

II.5.2. QoS evaluation

We use two techniques that help evaluate the calculated QoS value. The first one calculates an index of severity $IG_H(g)$ which indicates the severity of the case studied. It can be computed by approximating the integral of $f(t)$ where $f(t)$ is the function representing the state of the medical devices on a time interval $[t_i, t_f]$. The indicator has an objective; it compares the diabetic systems with the normal system.

II.5.2.1. Tester model

In the health field, an indicator is a single measure that captures a key dimension of health, such as how many people suffer from a heart attack. The international health community has come to a consensus regarding the key indicators representing health outputs and outcomes.

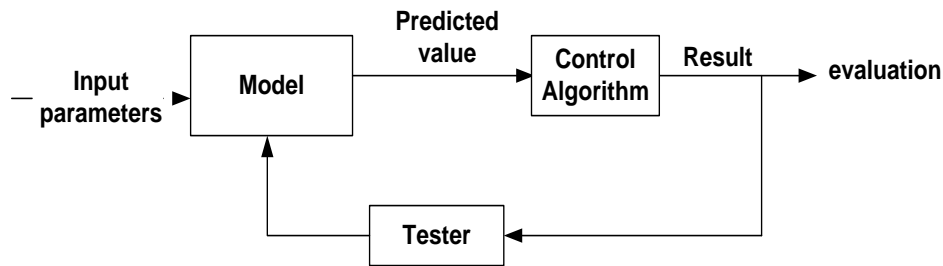


Fig. II. 9: Tester implementation using closed loop

The diagram of figure II.9 represents the third model sensors and actuators; it shows how medical devices can be interconnected to form a physiological closed-loop system. A high-level overview of the system allows creating many clinical cases that can benefit from closed-loop systems. To complete the closed loop we need a mathematical model, a control algorithm, and an effective tester. The model takes input parameters, then the predicted value is controlled by an algorithm to produce a values a results, the tester play the role of a feedback system that tests the result and send back the response to the model, in order to reach a better evaluation.

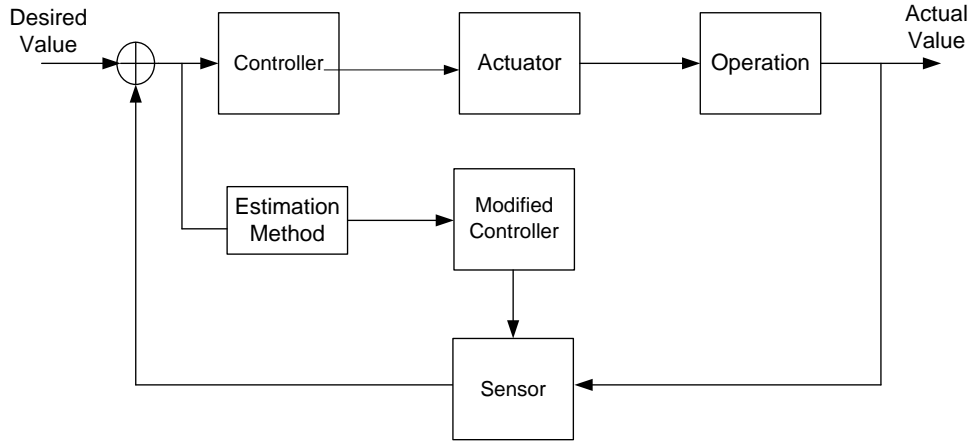


Fig. II. 10: Closed loop system overview.

Figure II.10 describes the closed- loop system and completes the previous one, where there is a need of a controller, sensor, actuator, an estimation function and the operation done. It's known that, we can enhance our closed-loop, for example by adding an estimation method that affects the control to be modified, and in relation with the sensor.

The QoS indicator has an objective. It compares the abnormal systems with the normal system. The graph of figure II.11 describes the simple indicator in an arbitrary case that we try to propose and apply it in the tester to help us reach our objective. We propose a new tester model in order to analyze the performance of all the components of the biomedical system.

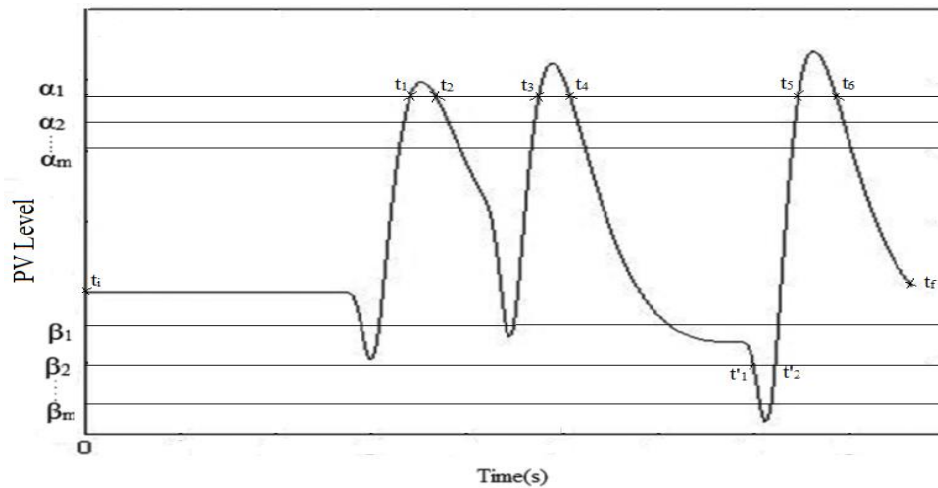


Fig. II. 11: Index of severity representation

We try to represent it using a mathematical representation, and to define a method to calculate a QoS value generated when applying it. Let us try to calculate the function $f(t)$ of the tester.

Consider $f(t)$ the function representing the *PV* controlled level on a time interval $[t_i, t_f]$. The parameters α_m and β_m are predefined constants / $\alpha_m = \{\alpha_1, \alpha_2, \alpha_3 \dots \alpha_n\}$ and $\beta_m = \{\beta_1, \beta_2, \beta_3 \dots \beta_n\}$. Set T_{Hyper} the duration in $[t_i, t_f]$ in the maximum value case and T_{Hypo} the duration in $[t_i, t_f]$ in the minimum value case.

For each t_j / $f'(t_j)=0$ and $f(t_j) > \alpha_m$:

$$\exists t_K, t_L (t_K < t_j < t_L) / f(t_K)=f(t_L)=\alpha_m \text{ and } T_{Hyper}=\sum(t_L - t_K) \quad (11)$$

For each t'_j / $f'(t'_j)=0$ et $f(t'_j) < \beta_m$:

$$\exists t'_K, t'_L (t'_K < t'_j < t'_L) / f(t'_K)=f(t'_L)=\beta_m \text{ and } T_{Hypo}=\sum(t'_L - t'_K) \quad (12)$$

Then, for a given simulation, there exists a value $g = \int_{t_k}^{t_f} f(t)$, which represents the surface bordered by the curve and the straight α_m or β_m . It exist an index of severity $IG_H(g)$ which indicates the severity of the patient's case. It can be computed by approximating the integral of $f(t)$, using Matlab built-in functions, as follows:

$$IG_H(g) = \text{trapz}(x_1, f') - \min(f')(\max(x_1) - \min(x_1)) \quad (13)$$

Where x_1 is the array of points of the formulated curve and f' is the array of values of *PV* level curve.

The changing in the parameters of the control algorithm and the values of g :

- Helps to apply many scenarios in order to have many results
- To deduce the best way to construct an effective tester.

II.5.2.2. Grid Analysis Representation

The error Grid can be designed to be used in diabetes screening, diagnosis, or to assess the accuracy of glucose monitoring. Trying to use a consensus error grid [49, 50, 51, 52], that helps to estimate the error of the performance of devices. The best representation used for sensors is for Clarke and Parkes in order to estimate the error of the performance of sensors, actuators, and controllers.

In 1987, Clarke and al. designed the error grid analysis (EGA), taking into consideration not only the difference between the system-generated and reference blood glucose values but also the clinical significance of this difference.

There are 5 risk categories are defined as follows: A: no effect on clinical action; B: little or no effect on clinical outcome; C: likely to affect clinical outcome; D: could have significant medical risk; and E: could have dangerous consequences.

In 2000, an updated version of the blood glucose error grid was proposed by Parkes [50] also present error grids as a way of specifying glucose performance needed for clinical purposes.

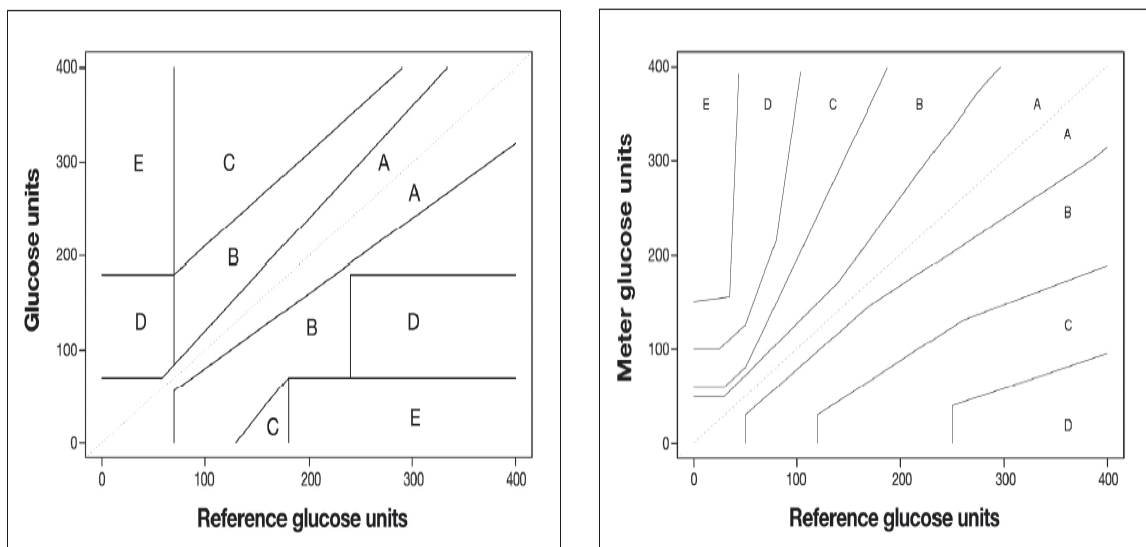


Fig. II. 12: Clarke and Parkes error grid for glucose.

In 2004, Clarke's group proposed continuous glucose–error grid analysis (CG-EGA) which had been specifically designed for evaluation of continuous glucose sensors. Clarke and colleagues and later Parkes and associates (figure II.12) presented error grids as a way of specifying glucose performance needed for clinical purposes. Clarke and Parkes grids are used to assess the accuracy of glucose monitoring [51]. In Figure II.12, using Clarke Grid, the "A Zone" is contiguous to a "D zone", which is considered a problem. This means that two results with almost the same amount of error could have very different clinical outcomes. In Parkes Grid, there exists a "B zone" as intervening between 'A zone' and any other higher zone. Analytical error represents the difference between tested glucose and reference method. The x-coordinate represent the accepted reference glucose value and y-coordinate the meter or tested glucose value.

Later in 2008, to visualize the overall glycemic control, a new tool has been introduced CVGA [52]. Using percentile, as a measurement unit instead of the absolute

minimum/maximum, reduces the vulnerability of the analysis to outliers. The grid is a simple representation in order to estimate the error of the performance of sensors, actuators, and controllers. The performance of the algorithm was analyzed with the consensus error grid using data sets generated by virtual patients and parameters changes (ref. chapter III)

The international Organization for Standardization (ISO) analytical accuracy standard 15197 for blood Glucose (BG) monitors specifies that 95% of data points must demonstrate acceptable, analytical accuracy but does not specify any performance targets for the remaining 5% of data points [53]. Analyze each simulation in purpose to have a graphical representation of the risk assessment for the patient due to choices on sensors, actuators.

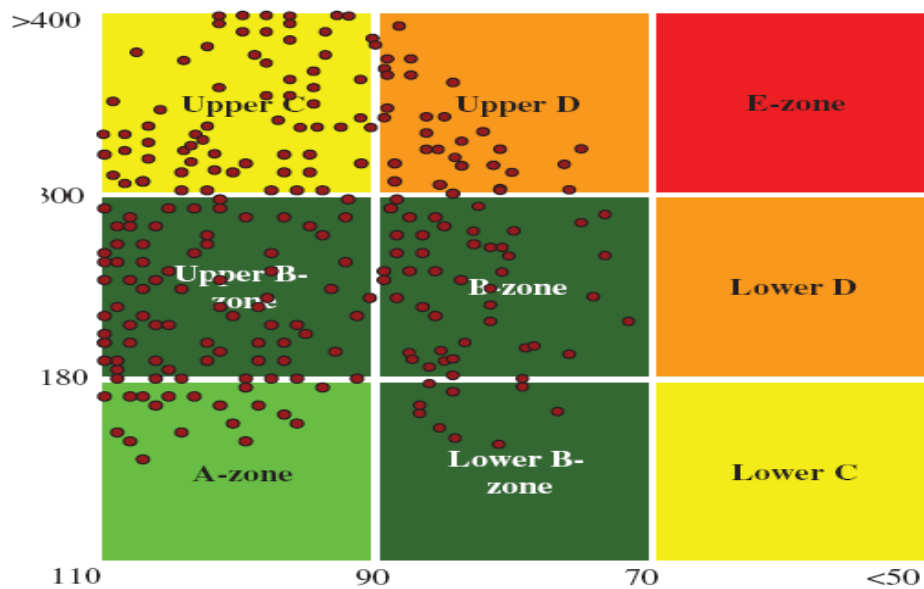


Fig. II. 13: Error grid graphic representation.

A point is plotted with x-coordinate the minimum BG and y-coordinate the maximum BG for an observation period. The plot is split into zones defined by their x- and y- coordinate ranges as follows:

Zone	Description
A-zone	Accurate control with x-range 110–90mg/dL and y-range 110–180mg/dL
B-zone	Benign control deviations with x=90–70mg/dL and y=180–300mg/dL
Lower B	Benign deviations into hypoglycemia: x=90–70mg/dL, y=110–180mg/dL
Upper B	Benign deviations into hyperglycemia: x=110–90mg/dL, y=180–300mg/dL
Lower C	Over-Correction of hyperglycemia: x<70mg/dL, y=110–180mg/dL
Upper C	Over-Correction of hypoglycemia: x=110–90mg/dL, y>300mg/dL

Lower D	Failure to Deal with hypoglycemia: $x < 70$ mg/dL, $y = 180\text{--}300$ mg/dL
Upper D	Failure to Deal with hyperglycemia: $x = 90\text{--}70$ mg/dL, $y > 300$ mg/dL
E-zone	Erroneous control: $x < 70$ mg/dL and $y > 300$ mg/dL

Table II. 1: Error grid zones definition.

Beginning with the index of severity tester representation and using a normalization function, we can represent the simulation as a point in order to construct a new error grid and then using it to evaluate the accuracy of glucose level measurements made by patients (figure II.14). In brief, each simulation is represented as a point in the grid; this point is calculated using normalization function. This function takes as input the array of the simulation and the percentile value, and gives as output a point in a grid using 97.5th and 2.5th percentile.

The 2.5th and 97.5th percentiles represents a single direction between two test boxes in the best and worst scenarios, respectively, and the median (50th percentile) would indicate a normal value between two measurements points. If you have a set of values, then the ones in the 97.5th percentile are the values greater than 97.5% of the others.

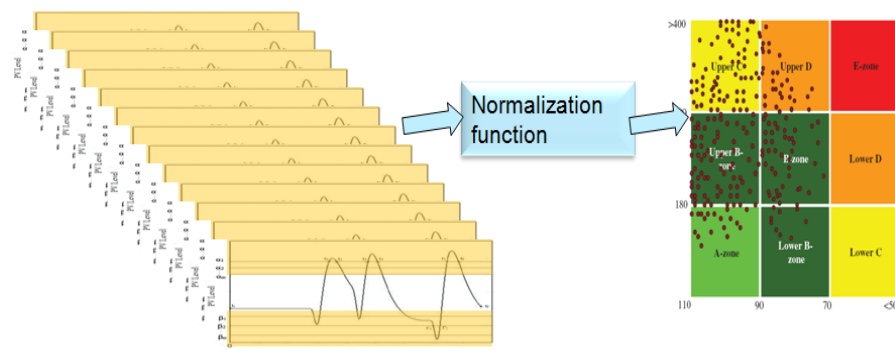


Fig. II. 14: Graphic representation using normalization function.

Having many simulations scenarios help to plot them as points in this grid. The background squares behind zones in the tester presentation can represent the unwanted states. The normalization function can be percentile or any measurement unit that help to calculate a point.

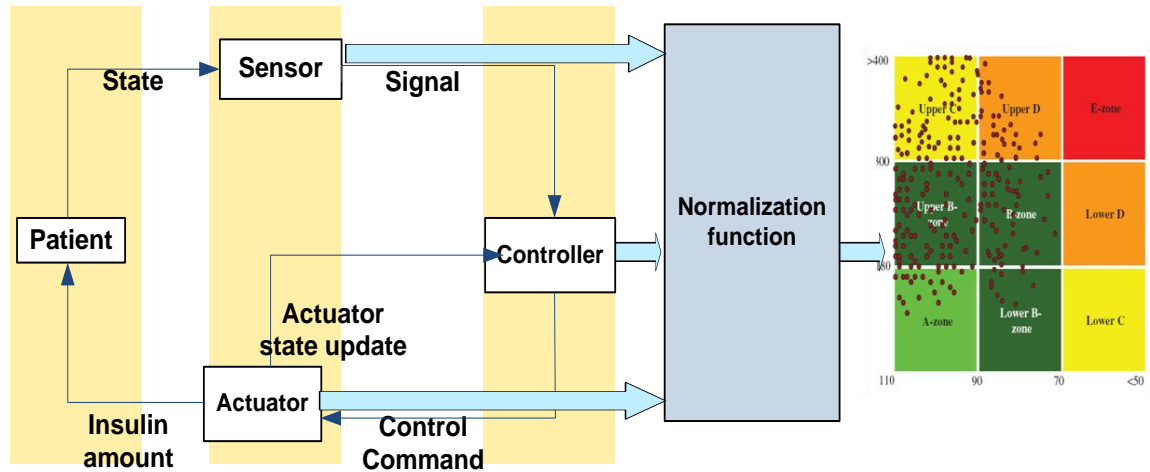


Fig. II. 15: Simulate the performance of equipments.

In this work, the Grid representation will help not only to simulate the performance of sensors but also actuators and controllers (figure II.15). Analyzing the performance of the biomedical devices helps to identify the usage of such sensors, actuators or controllers. In brief, using simulation of the following components: sensors, actuator or controller, and a normalization function, we can represent this simulation using grid analysis.

II.6. Conclusion

We have provided a global flexible architecture of the human body model with mathematical representation. A full simulation representation of the simulator components, analyzer part and the QoS indicator schema and the evaluation method has been also presented. The purpose of this architecture is to have a complete environment with the ability to simulate medical equipments, and test their performance.

From these results, the tester that we are creating is a robust tool that will improve the testing in biomedical fields without having any risk on patients. The plot of each simulation as a point in a grid and then split it into zones defined by their x- and y-coordinate ranges, help to construct a new error grid and use it to evaluate the accuracy of glucose level measurements made by the patients.

Chapter III: Case study - Systems implementation

III.1.Introduction

Embedded devices used in medical research help to improve the quality of diagnostic tools available to doctors as well as the treatments offered to patients. Recently, embedded technologies have become very important for future manufacturing enterprises as well as in industrial engineering. Advanced researches in microelectronics and software engineering allow many techniques to be implemented in embedded medicals devices [54]. Biomedical research seeks all times a good reasoning for solving medical problems based on intensive work and great debate. It often deals with theories that have been proven after observations or experiments [55, 56].

In such system there is a necessity to understand the effect of insulin and carbohydrates on blood glucose evolution for a specific patient. In recent years, many mathematical equations or known as mathematical models have been used to create simulators to test different types of treatment and have showed many control approaches to automatic regulation of blood glucose [57]. These models arrive to become a choice for control algorithm. Better understand this system via mathematical equations helping to simulate it in normal life conditions which can be useful in diabetes research. Model the system, and know how it interacts with the environment.

In last decades, simulation models of the glucose-insulin control system during meals and normal daily life has been proposed for studying the pathophysiology of diabetes [58, 59]. Simulation experiments with the mathematical model of a system are valuable tools for student education and medical fields [60].

In this chapter, we implement a mathematical model applicable in our prototype that was described in chapter II, and the *PV* controlled is for example the glucose level in the blood. The event starts by eating several sweet candies. In the digestive system, they are rapidly degraded in various simple, which causes a rapid rise in blood level. The increased glucose level stimulates pancreatic cells responsible of produces the insulin, which then release it into the blood. Insulin accelerates the uptake of glucose by most cells and promotes its storage as

glycogen in the liver and muscles; the body in somehow set glucose in reserve. Therefore glucose returns to normal reference value and the event that triggered insulin secretion also decreases. Glucagon, another pancreatic hormone, has an opposite effect. It is released when glucose levels fall below the reference value.

We begin by the glucose insulin system and create a new framework in order to test the performance of all system components. The framework consists of simulating a mathematical model of human body in order to implement in a microcontroller, developing a control algorithm for the model and applying parametric models of activities to show how medical devices can be interconnected to form a physiological closed-loop system. This chapter deals with the simulation set-up, the explanation of the modeling work and the presentation of the control algorithm. We also present the three models implementations which are the model of human body, the model of artificial pancreas and the model of sensors and actuators.

We propose a new model for global simulating of biomedical equipments (including human interaction models). We begins by an in-silico study for type 1 diabetes mellitus patients using a mathematical model, with implementation of a our control algorithm. This model was developed to operate in the closed loop of the glucose insulin. Next, the model of artificial pancreas has been implemented to control the mathematical model of human body.

III.2. Glucose–Insulin System

A relationship exists between the different organs of the body: liver, muscle, pancreas... in order to formulate a glucose-insulin system, and to adjust the concentration of glucose level in the blood (figure III.1). The glucose-insulin system within the human body acts normally as a regulator of the glucose concentration in the blood (BG), thus preventing what called in medical terms for high blood glucose (hyperglycemia) or low glucose level (hypoglycemia). The glucose-insulin system is an example of a closed-loop physiological system. The normal regulation of the blood glucose level is achieved by the glucose–insulin system. A healthy person normally has a fasting sugar level in the range of 70–110 mg/dL.

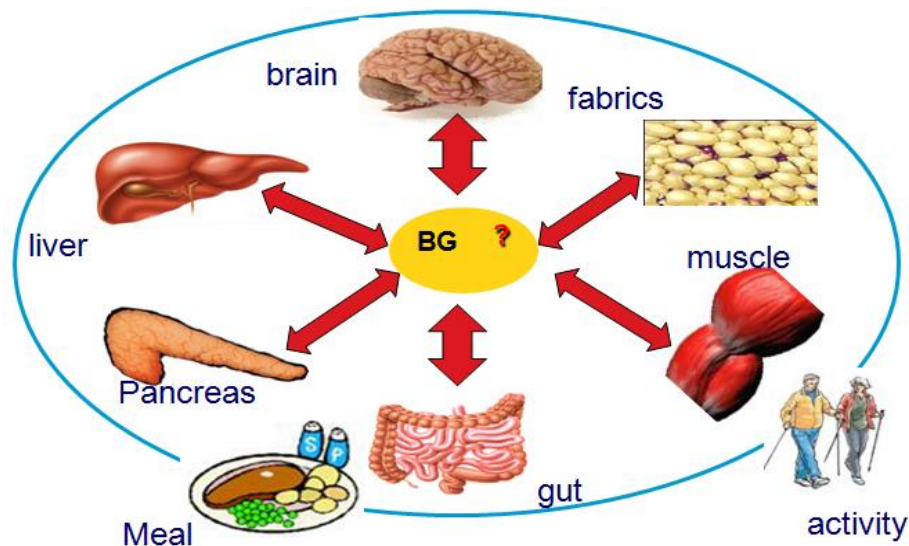


Fig. III. 1: The blood glucose–insulin system.

The blood glucose level should be maintained in a very narrow range; insulin and glucagon, secreted from the pancreas, are the hormones that regulate this level. When the control of insulin levels fails, diabetes mellitus will result. Insulin is a hormone produced by the β -cells of the islets of Langerhans in the pancreas. A high insulin level promotes storage of glucose, and a low insulin level signals the need for the release of glucose fuels, currently in storage, back into the blood stream. Glucagon and adrenaline signals the liver to release glucose. Too much glucose removal from the blood-stream can result in dangerously low blood glucose levels. Glucagon and insulin are part of a feedback system that maintains the blood glucose at the correct level. For example in case of hypoglycemia, the α -cells react by

releasing glucagon, which acts on the liver cells, causing them to release glucose into the blood until the person is back in the green area again

In such system there is a need to understand the effect of insulin and carbohydrates on blood glucose evolution for a specific patient. This is done through consideration of the glucose-insulin system, its inputs, outputs and modeling it. In this case, we will use these parameters to test biomedical equipment without the use of real patients.

The integration between the human system and the electronic system to act as one system is shown in figure III.2. The glucose-insulin system with a monitoring device that take the glucose level and send signal to the controller, the controller communicate with the actuator by sending control command, the actuator send the amount of insulin regulated by the controller. In brief, this integrated system is composed from patient, sensor for monitoring, controller and an actuator. The interaction between them is described in details below in section III.2.3. The objective is to simulate the whole system of this co-simulation, in other words, build a complete system with the ability to simulate it and detect the performance of all system components.

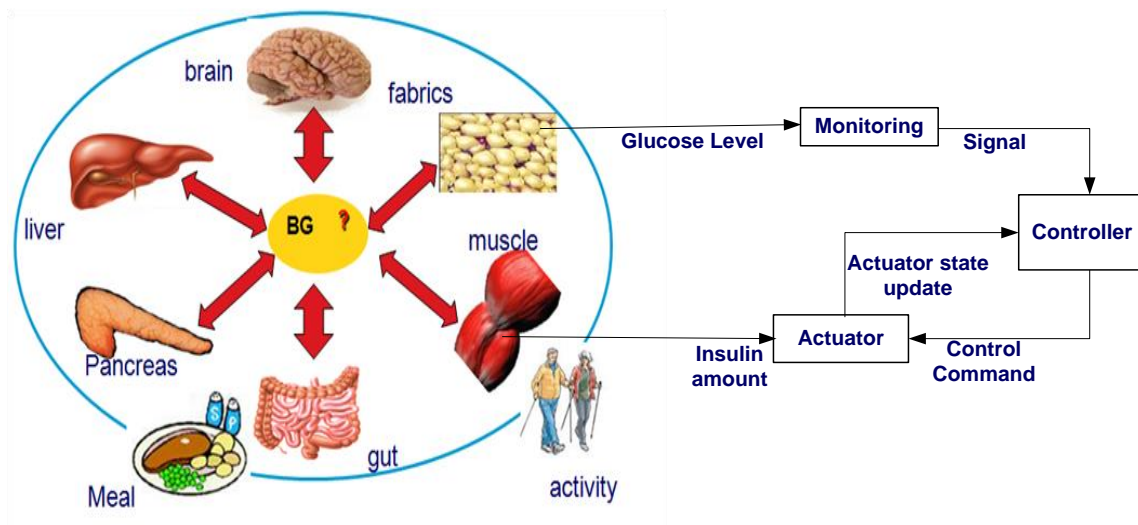


Fig. III. 2: Interaction schema of the integrated system.

The objective is to formulate a co-simulation of physiological and embedded system models (codes, sensors, actuators ...), and schematization of the natural regulation of glucose in the form of mathematical models

III.2.1. Diabetes mellitus overview

Diabetes is often described as a chronic illness of carbohydrate metabolism, characterized by high or low blood sugar level. In other term, this disease is defined by the inability of the body to produce or properly use insulin. It can be associated with serious complications and premature death but it can be controlled by taking measures that lower the risk of complications. Diabetes results from many causes, though genetics and environmental factors such as obesity and lack of exercise appear to play a major role. Over time, the number of β -cells starts to decline, and then the type 2 diabetic patient must be treated with insulin injections like the type 1 diabetic to maintain his/her blood sugar at normal levels.

Its known when a healthy patient eats a meal, the carbohydrates are broken down into glucose, galactose and fructose, with galactose and fructose transformed quickly into glucose. During this period insulin level increases naturally to stimulate glucose uptake. Insulin increase results in increased glucose uptake by liver and peripheral tissues, keeping plasma glucose level within normal range. Unlike a diabetic person, insulin effect on glucose regulation strongly depends on the quality of insulin therapy, depending on the insulin amount administered and the time of administration. For this reason, any late in insulin administration would result in hyperglycemia at the beginning of the meal and hypoglycemia at the end of the meal or shortly after. There are three types of diabetes [61]:

- **Type 1 diabetes:** results from the body's failure to produce insulin, and presently requires insulin injection. (IDDM for short or juvenile diabetes). Without insulin, glucose remains in the bloodstream, so blood glucose levels increase, especially after meals are consumed. The glucose is then passed out of the body in the urine. Today the treatment of this type of diabetes is done by injecting insulin into the body, by exercising and keeping a healthy diet.
- **Type 2 diabetes:** results from insulin resistance, a condition in which cells fail to use insulin properly, sometimes combined with an absolute insulin deficiency. Non-insulin-dependent diabetes mellitus (NIDDM for short or adult-onset diabetes). It is considered as the most common type of diabetes.

Over time, the number of β -cells starts to decline, and then the type 2 diabetic patient must be treated with insulin injections like the type 1 diabetic to maintain his/her blood sugar at normal levels.

- **Gestational diabetes:** is when pregnant women, who have never had diabetes before, have a high blood glucose level during pregnancy. It may precede development of type 2 DM.

All types of diabetes are treatable by different ways, such as tablets, regular insulin injections, as well as a special diet and exercise.

III.2.2. Glucose Level

The human body uses a complex metabolic system to sustain life and power its everyday actions. It converts complex forms of food into glucose, a type of sugar used in energy expenditure. A simple definition of Glucose level is the amount of glucose in the blood, this level vary before and after meals, and at various times of day.

Diabetics have to be more or less constantly aware of the current concentration of blood glucose. Daily measurement can be carried out using an electronic glucometer [62]. In order to prevent the complications associated with diabetes mellitus, we must maintain the blood glucose level near normal range. The following table III.1 gives an approximate interpretation of the major blood glucose recorded levels.

mg/dl	Interpretation
35	Extremely low
55	Low
75	Slightly low
100	Normal
90-110	Normal pre-prandial in non-diabetics
150	Normal postprandial in non-diabetics
180	Maximum postprandial in non-diabetics
200	A little high
270	A little high to very high depending on patient
300	Arrive to a sensitive state
360	Getting up there

400	Max mg/dl for some meters and strips
600	High danger of severe electrolyte imbalance

Table III. 1: Interpretation and approximate value of BG level.

According to table III.1, it's clear that range between optimal state and hypoglycemia is very narrow. Since hypoglycemia is definitely the worst state to be in, it should be obvious why many diabetics prefer to “overdose” themselves with sugar. This is one of the reasons why it's so hard, especially for type 1 diabetics, to specify the right amount of insulin to be injected; even a small miscalculation could have unpleasant consequences.

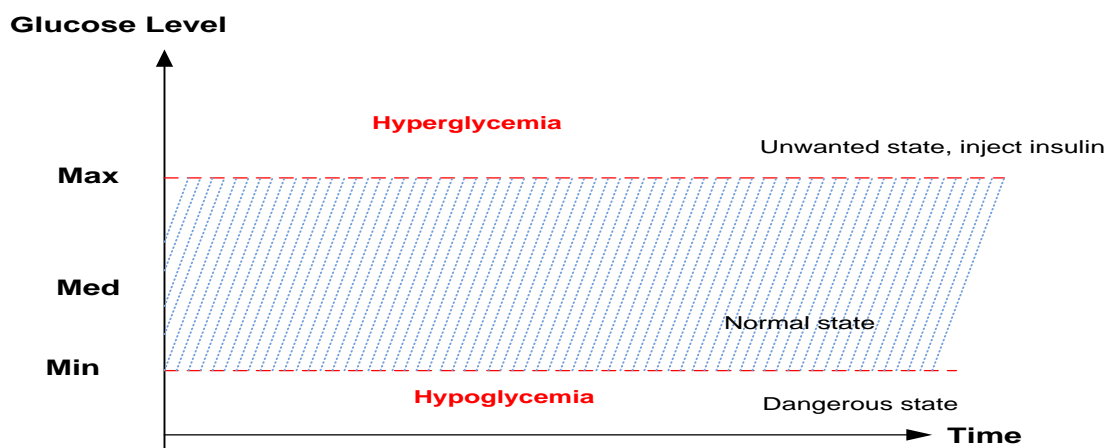


Fig. III. 3: Blood glucose level presentation.

We can divide the level of sugar in three states (figure III.3): dangerous, normal, unwanted. The dangerous range where we have a very low level of blood sugar, the part can be considered as normal as people without diabetes and unwanted when you have a broad level of high blood sugar.

III.2.3. Closed-loop

The closed-loop system use the feedback from the output to completes its operating cycle within the system. A closed-loop control system, also known as a feedback control system is an open-loop with feedback, which means that the output value is returned to the input in order to improve its quality. It's important in closed-loop control system, to measure the

controlled variable and calculate the control error value, in order to arrive to remove or reduce this error.

Basically, blood glucose sensors and an insulin pump help to control insulin delivery system [63, 64]. In brief, an implanted blood glucose sensor measure glucose level so an insulin pump, attached to a patient's body, continuously inject insulin into it. If the systems are interconnected in a cycle, we have a closed loop system. We give a high-level overview of the system, and this allows creating many clinical cases that can benefit from closed-loop systems. Figure III.4 show how a sensor detect the state of a patient, then a signal is sent to the controller, the controller upon an algorithm decide to send a controller command to the actuator, the actuator play his role to send or not the necessary insulin amount. The state of the actuator is always send to the actuator, to let him know if last command is activated or not, and the amount of insulin in the reservoir.

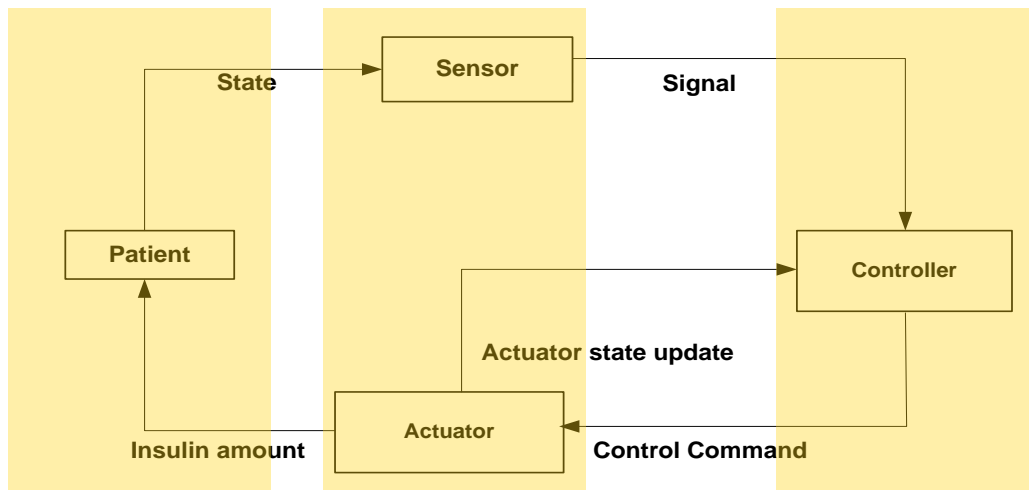


Fig. III. 4: Closed Loop system graph.

The closed-loop system can be tested in many levels by implementation of test scenarios based on parametric models of activities, taking meals and setting of the patient (weight, pathology ...). The background squares behind zones mark these levels. These levels are detailed as follow: level Patient where different parameters can represent a set of range test, the second level of sensor and actuators in case where sensor and actuators are working fine or any problem in sensor or actuator give corruption in glucose level that should affect the system, level of controller where must of control testing can be adjusted and modified to arrive to our objective.

The closed-loop system completes its operating cycle within the system and no external interaction to diabetic patients is required. In other words, the closed-loop control uses the feedback from the output.

III.2.4. Mathematical models

Embedded systems used in medical care help more and more to improve the quality of diagnostic tools available to doctors as well as treatments offered to patients. The first step in this process resides in a better understanding of the glucose-insulin system via mathematical equations: modeling the system and knowing how it interacts with the patient's environment will help to simulate it in normal life conditions and can be thus useful in diabetes research.

Since the sixties, many mathematical models have been developed to better understand the glucose insulin regulatory system. Various models have been proposed to describe the short-term glucose-insulin dynamics. A physiological model that captures the glucose-insulin system dynamics is thus the basis for more optimally addressing the glycemic control problem. In brief, metabolic modeling of the glucose-insulin system has a very deep history in the published literature. The vast majority of these models have their roots in basic compartment modeling with differential equations (Appendix A). To date, the primary use of metabolic models has been the development of model-based measures to assess metabolic parameters, with a particular focus on measuring insulin sensitivity. Mathematical models can be used to create simulators to test different types of treatment.

a. Bergman Minimal model

Minimal model with low-order was for estimation of insulin sensitivity and glucose effectiveness. This model was developed by Richard N. Bergman and therefore is called Bergman's minimal model [65, 66]. It is commonly used to analyze the results of glucose tolerance tests in humans and laboratory animals. Bergman's minimal model describes the body as one compartment model, which mean as a compartment/tank with a basal concentration of glucose and insulin. It's a three-compartment minimal model to analyze the glucose disappearance and insulin sensitivity during an intravenous glucose tolerance test. Modifications have been made to the original minimal model to incorporate various physiological effects of glucose and insulin.

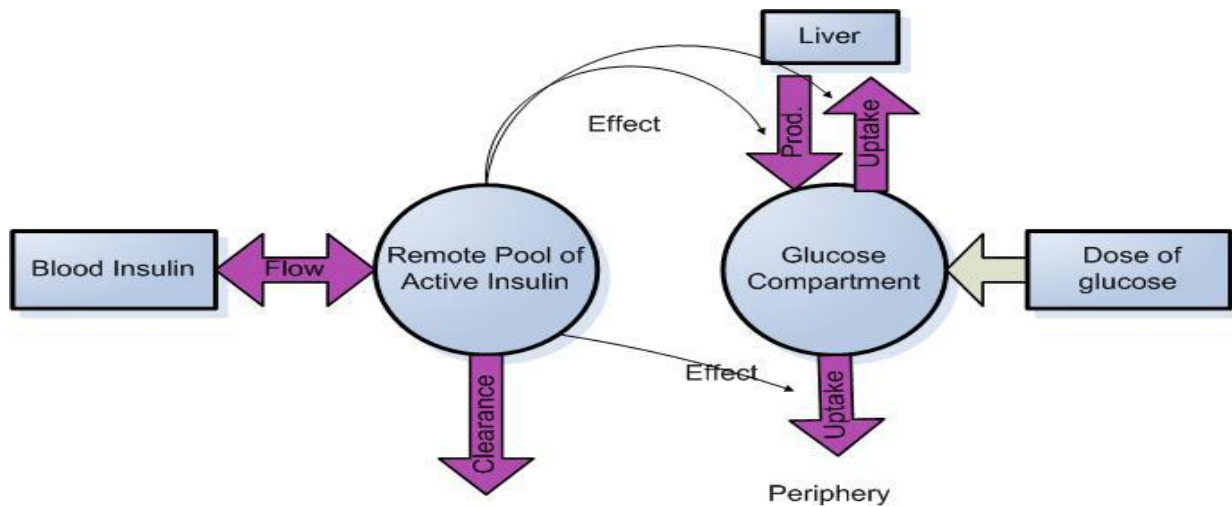


Fig. III. 5: Minimal glucose model.

As seen in the picture above (figure III.5), the glucose flows in and out of this compartment at a steady rate, resulting in a basal concentration. Bergman's minimal model consists of a glucose compartment, a remote insulin compartment and a plasma insulin compartment. Glucose uptake is influenced by plasma insulin through a remote compartment. This is a schematic of the Bergman minimal model. The minimal model, compatible with some known physiological facts, can simulate the glucose-insulin system with minimal identifiable parameters and is computationally suitable for parameter estimation and real-time control.

b. Hovorka model

This model was developed primarily by Roman Hovorka and is therefore being referenced as Hovorka's model [67, 68, 69]. It has two inputs, meal disturbances and insulin infusions and simulates a person with type 1 diabetes. It extends the original minimal model by adding three glucose and insulin sub-compartments in order to capture absorption, distribution, and disposal dynamics, respectively.

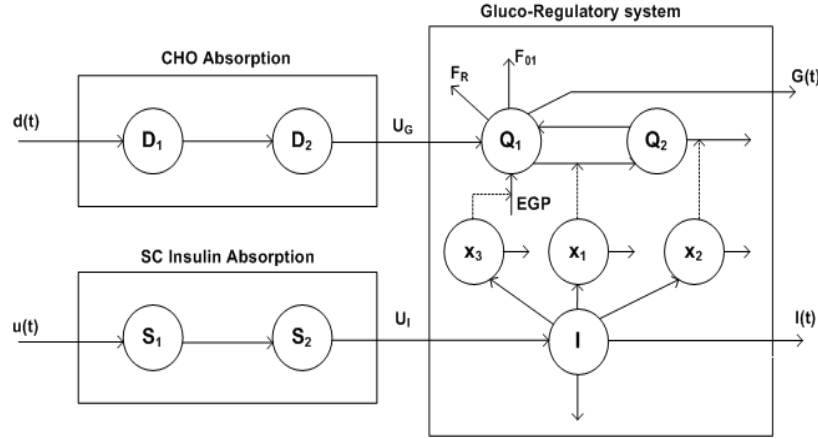


Fig. III. 6: Hovorka model.

This is a more complete model for this system (figure III.6). It was developed using glucose tracers. The main idea of this model is to divide the compartments in specific biological processes.

The parameters of the system are defined as follows: Q_1 and Q_2 represent the masses of glucose in the accessible and non-accessible compartments; F_{01} is the total non-insulin-dependent glucose flux corrected for the ambient glucose concentration and F_R is the renal glucose clearance; G is the measurable glucose concentration; EGP represents endogenous glucose production; x_1 , x_2 , and x_3 represent three actions of insulin on glucose kinetics; S_1 and S_2 are a two-compartment chain representing absorption of subcutaneously administered short-acting; I describes the plasma insulin concentration; D_1 and D_2 are a two-compartment chain representing the amount of carbohydrates digested. BW is the weight of the patient.

The equations of the model are defined below:

$$\frac{dQ_1(t)}{dt} = U_G(t) - F_{01,c} - F_R(t) - x_1(t)Q_1(t) + K_{12}Q_2(t) + EGP_0(1 - x_3(t))$$

$$\frac{dQ_2(t)}{dt} = x_1Q_1(t) - (K_{12} + x_2(t))Q_2(t)$$

$$\frac{dS_1(t)}{dt} = u(t) - \frac{S_1(t)}{T_s}$$

$$\frac{dS_2(t)}{dt} = \frac{S_1(t)}{T_s} - \frac{S_2(t)}{T_s}$$

$$D_1 = D_1 + h * \frac{dD_1(t)}{dt} \quad / \quad h \text{ constant variation}$$

$$D_2 = D_2 + h * \frac{dD_2(t)}{dt}$$

The insulin absorption rate, $U_I(t)$ (mU/min) can be calculated:

$$U_I(t) = \frac{S_2(t)}{T_S}$$

The insulin concentration, $I(t)$ (mU/L), is found by solving the following differential equation:

$$\frac{dI(t)}{dt} = \frac{U_I(t)}{V_I} - K_e I(t)$$

x_1 , x_2 and x_3 are calculated using the following three differential equations, depending on only the plasma insulin concentration and parameters

$$\frac{dx_1(t)}{dt} = -K_{a1}x_1(t) + K_{b1}I(t) \quad / \quad k_{b1} = S_{IT} k_{a1}$$

$$\frac{dx_2(t)}{dt} = -K_{a2}x_2(t) + K_{b2}I(t) \quad / \quad k_{b2} = S_{ID} k_{a2}$$

$$\frac{dx_3(t)}{dt} = -K_{a3}x_3(t) + K_{b3}I(t) \quad / \quad k_{b3} = S_{IE} k_{a3}$$

Symbol	Value /Unit	Description
K_{12}	0.066 min ⁻¹	Transfer rate
K_{a1}	0.006 min ⁻¹	Deactivation rate
K_{a2}	0.06 min ⁻¹	Deactivation rate
K_{a3}	0.03 min ⁻¹	Deactivation rate
K_e	0.138 min ⁻¹	Insulin elimination rate
T_D	40 min	CHO absorption constant
T_S	55 min	Insulin absorption constant
A_G	0.8	CHO to glucose utilization
EGP_0/BW	0.0161 mmol.Kg ⁻¹ min ⁻¹	Liver glucose production at zero insulin
F_{01}/BW	0.00097 mmol.Kg ⁻¹ min ⁻¹	Insulin independent CNS consumption
S_{IT}	51.2e ⁻⁴ L/mU	Insulin sensitivity of transport/ distribution
S_{ID}	8.2e ⁻⁴ L/mU	Insulin sensitivity of disposal
S_{IE}	520e ⁻⁴ L/mU	Insulin sensitivity of EGP

Table III. 2: Hovorka model parameters.

c. Man-Rizza-Cobelli's Model

This model was developed by Chiara Dalla Man, Robert A. Rizza and Claudio Cobelli, therefore, called Man-Rizza Cobelli model [70] (figure III.7). They provided a

revised minimal model in order to separate the effects of glucose production from utilization. They proposed a nonlinear model consisting of glucose, insulin and glucagon subsystems. Picture below describes this model by giving a general overview of the glucose-insulin control system. Glucose and glucagon subsystem were modeled using a single compartment respectively, and insulin subsystem was expressed as a five-compartment model.

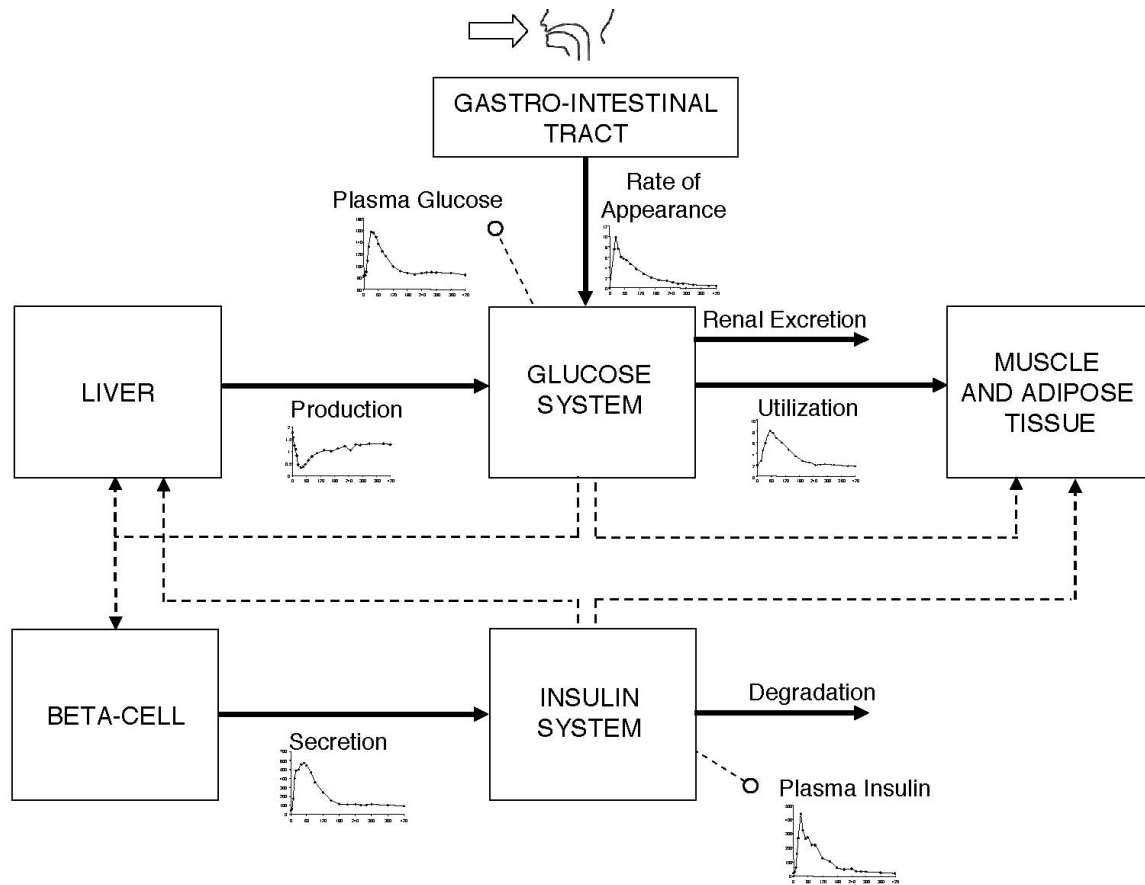


Fig. III. 7: Man-Rizza-Cobelli model.

The model is made up of a glucose and insulin subsystem linked by the control of glucose on insulin secretion and by insulin on glucose utilization and endogenous production. The glucose subsystem consists of a two-compartment model of glucose kinetics: insulin-independent utilization occurs in the first compartment, representing plasma and fast equilibrating tissue, while insulin-dependent utilization occurs in a remote compartment, representing peripheral tissues. The insulin subsystem also consists of two compartments, the first representing the liver and the second the plasma.

d. Comparison and decision

Bergman's minimal model provides a good approximation of the system, but omits several important physiological functions and features of insulin, which are included in the

model of Hovorka. It includes most of the physiological parameters of the glucose and insulin action. In Hovorka, features non-insulin-dependent has been smoothed to avoid discontinuities in the system. Man-Rizza-Cobelli model is a bit more complex than Hovorka's, and is supposed to be more realistic. Like Hovorka this simulation model is based on the principle of having a system of differential equations, and we have several systems (organs), which is all modeled separately and all connected.

These models were intended for different purposes:

- Measurement of insulin sensitivity and control (Bergman)
- Simulation and control (Hovorka)
- Simulation (Man-Rizza-Cobelli)
- Control (Panunzi)

We have the choices to choose between Hovorka and Cobelli models to implement it to our system, noting that these two models have good presentation of the human system body, so as example we implement the Hovorka model. Noting that, the success in applying one of them using embedded c language, will allow us to apply any other one.

III.3.Models Implementation

A simulator is developed to make virtual experiments that show the effects of changes parameters on glucose levels in insulin-dependent diabetes mellitus. We need to perform reachability/safety analysis of the system.

In this part of this chapter we create a new framework in order to test the performance of all system components. The framework begins by simulating a mathematical model of the human body. This model was developed to operate in the closed loop of the glucose insulin. Next, the model of artificial pancreas has been implemented to control the mathematical model of human body. Finally a new tester model was created in order to analyze the performance of all the components of the glucose-insulin system. The Keil debugger file play the role of the patient and the programmed algorithm play the role of the controller.

III.3.1. Implementation of Human body

Mathematical models of glucose regulation have been studied over years. Recently, modeling an insulin-glucose system has been presented [71].

This section provides an overview of the model while references concerning the

mathematical equations are defined in previously in section III.2.4.d. The model used has been selected according to the need of the research. This is a complete model for the glucose-insulin system during a meal and it was developed using glucose tracers. The main idea of this model is to divide the compartments in specific biological processes.

First of all, we have validated the model of human body using “Hovorka” that we have implemented using the reference scenario find in [68, 69]. We have implemented it using Keil in an embedded code program, to act as a patient. In this scenario insulin is taken before the meal. We consider meals, taken by a person, which represents breakfast, lunch and dinner in a normal day. Just as a point of reference we use $CHO_1 = 45g$ and $CHO_2 = CHO_3 = 70g$, which means that we assume that the amount of carbohydrates eaten for breakfast is 45g, and so on. The insulin doses are also exactly the same with 2U/L for breakfast and 3U/L for lunch and dinner. The Hovorka model is implemented based on algorithms defined from table III.3 to table III.8.

Table III.3 and table III.4 represents the definition and implementation of the function Hovorka in keil. This function takes three inputs (the weight, the insulin absorption rate and the meal input, and describes how we implement the mathematical equations of Hovorka.

Function name	Hovorka
Function prototype	Function void Hovorka(float we,float uI, float dbas)
Behavior description	Hovorka model simulation
Input	we: weight uI: the insulin absorption rate dbas: the meal input in mmol/min
Output	None

Table III. 3: Hovorka Function definition.

Algorithm: Hovorka model applied in Keil debugger file
Input : Three decimal variables : weight decimal insulin absorption rate decimal meal input decimal Variables: Declare all variables as decimal values Begin

```

Call function testGB with weight as parameter
dD1  $\leftarrow A\_G * dbas - D1 / \tau\_G$ 
dD2  $\leftarrow D1 / \tau\_G - D2 / \tau\_G$ 
dS1  $\leftarrow uins - S1 / \tau\_I$ 
dS2  $\leftarrow S1 / \tau\_I - S2 / \tau\_I$ 
dQ1  $\leftarrow -(F\_01c + F\_R) - x1 * Q1 + k\_12 * Q2 + U\_G + EGP\_0 * (1 - x3)$ 
dQ2  $\leftarrow x1 * Q1 - (k\_12 + x2) * Q2$ 
dI  $\leftarrow U\_I / V\_I - k\_e * I$ ;
dx1  $\leftarrow k\_b1 * I - k\_a1 * x1$ 
dx2  $\leftarrow k\_b2 * I - k\_a2 * x2$ 
dx3  $\leftarrow k\_b3 * I - k\_a3 * x3$ 
D1  $\leftarrow D1 + h * dD1$ 
D2  $\leftarrow D2 + h * dD2$ 
S1  $\leftarrow S1 + h * dS1$ 
S2  $\leftarrow S2 + h * dS2$ 
Q1  $\leftarrow Q1 + h * dQ1$ 
Q2  $\leftarrow Q2 + h * dQ2$ 
I  $\leftarrow I + h * dI$ 
x1  $\leftarrow x1 + h * dx1$ 
x2  $\leftarrow x2 + h * dx2$ 
x3  $\leftarrow x3 + h * dx3$ 
End

```

Table III. 4: Hovorka function implementation algorithm (1).

Table III.5 and table III.6 describes the definition and the implementation algorithm of the function used to calculate the values of parameters related to patient by the weight. It's called the function of metabolic processes and takes as input the weight.

Function name	testGB
Function prototype	Function void testGB (long bwi)
Behavior description	Hovorka model simulation, called by Hovorka function to calculate specific parameters related to weight. It's the function of metabolic processes.

Input	bwi: weight
Output	None

Table III. 5: TestGB function definition.

Algorithm: Hovorka model applied in Keil debugger file
<p>Variables: Declare all variables as decimal values</p> <p>Input : one decimal variable represent the weight :</p> <p style="padding-left: 40px;">bwi decimal</p> <p>Begin :</p> <p>BW \leftarrow bwi</p> <p>U_G \leftarrow D2/tau_G</p> <p>U_I \leftarrow S2/tau_I</p> <p>V_I \leftarrow 0.12*BW</p> <p>V_G \leftarrow 0.16*BW</p> <p>F_01 \leftarrow 0.0097*BW</p> <p>EGP_0 \leftarrow 0.0161*BW</p> <p>uins \leftarrow 0.0954119*BW</p> <p>Gb \leftarrow Q1/V_G</p> <p>If Gb \geq 4.5 then</p> <p style="padding-left: 40px;">F_01c \leftarrow F_01</p> <p>else</p> <p style="padding-left: 40px;">F_01c \leftarrow F_01*Gb/4.5</p> <p>End if</p> <p>If Gb \geq 9 then</p> <p style="padding-left: 40px;">F_R \leftarrow 0.003*(Gb - 9)*V_G</p> <p>else</p> <p style="padding-left: 40px;">F_R \leftarrow 0</p> <p>End if</p> <p>End</p>

Table III. 6: Hovorka function implementation algorithm (2)

Table III.7 and table III.8 describes the definition and the implementation algorithm of the simulation of the Hovorka model in keil. We have simulated a whole day (more details in Appendix B). We simulate a day life of a patient using Hovorka model, where the patient

takes insulin before each meal. The simulation progress is for a diabetic person that tries to maintain his glucose level within limit. Before each meal (breakfast, lunch and dinner), the system call a function to get necessary insulin shot needed, and during it, the system call a function to calculate the meal input. A function named *RelevantMoments* is used to specify the time of each meal, for example at 8.00 breakfast, at 12.00 lunch, and at 19.00 dinner.

Function name	Testins
Function prototype	Func void Testins ()
Behavior description	This function is a day life scenario for a patient. The simulation started at 7.00, for example, then these times would simulate at 8.00 breakfast, 12.00 lunch, and 19.00 dinner.
Input	bwi: weight
Output	None

Table III. 7: Testins function definition.

Algorithm: Day life scenario algorithm for function Testins.
<p>Variables: Declare all variables as decimal values</p> <p style="padding-left: 40px;"><i>bCHO</i>, <i>lCHO</i>, <i>dCHO</i> are quantities eaten during each meal</p> <p style="padding-left: 40px;"><i>eatingTime</i> represent the time needed for eating a meal</p> <p>Begin :</p> <p style="padding-left: 20px;">Call the function <i>RELEVANTMOMENTS</i> to specify the time of each meal</p> <p style="padding-left: 20px;">Declare <i>countspan</i> as time during simulation</p> <p style="padding-left: 20px;">Step 1 : initialize the value of Hovorka models</p> <p style="padding-left: 20px;">Step 2: Midnight</p> <p style="padding-left: 20px;">If <i>countspan</i> is during midnight then</p> <p style="padding-left: 40px;">Call function Hovorka with parameters Weight , insulin shot</p> <p style="padding-left: 20px;">End if</p> <p style="padding-left: 20px;">Step 3: Before breakfast</p> <p style="padding-left: 20px;">If <i>countspan</i> is before breakfast then</p> <p style="padding-left: 40px;">Call function to get insulin shot needed before meal</p> <p style="padding-left: 40px;">Call function Hovorka with parameters Weight , insulin shot</p> <p style="padding-left: 20px;">End if</p> <p style="padding-left: 20px;">Step 4: Breakfast start</p>

If *countspan* is during breakfast **then**

Calculate the meal input *dch*

$dch \leftarrow bCHO / (180 * eatingTime) * 1000$

Call function Hovorka with parameters Weight , insulin shot, meal input

End if

Step 5: Breakfast stop

If *countspan* is after breakfast **then**

Call function Hovorka with parameters Weight , insulin shot

End if

Step 6: Before lunch

If *countspan* is before lunch **then**

Call function to get insulin shot needed before lunch meal

Call function Hovorka with parameters Weight , insulin shot

End if

Step 7: lunch start

If *countspan* is during lunch **then**

Calculate the meal input *dch*

$dch \leftarrow lCHO / (180 * eatingTime) * 1000$

Call function Hovorka with parameters Weight , insulin shot, meal input

End if

Step 8: lunch stop

If *countspan* is after lunch **then**

Call function Hovorka with parameters Weight , insulin shot

End if

Step 9: before dinner

If *countspan* is before dinner **then**

Call function to get insulin shot needed before dinner meal

Call function Hovorka with parameters Weight , insulin shot

End if

Step 10: dinner start

If *countspan* is during dinner **then**

Calculate the meal input *dch*

$dch \leftarrow dCHO / (180 * eatingTime) * 1000$

Call function Hovorka with parameters Weight , insulin shot, meal input

End if

Step 11: dinner stop

```

If countspan is after dinner then
    Call function Hovorka with parameters Weight , insulin shot
End if
Step 12: 24 hours passed
If hours =24 then
    Exit
End if
End

```

Table III. 8: Hovorka scenario implementation algorithm.

The graph below (figure III.8) represents this case, we notice that the glucose level is smooth and its range is inside the limit at all times. This scenario corresponds to the usual insulin injection made by a well educated diabetic.

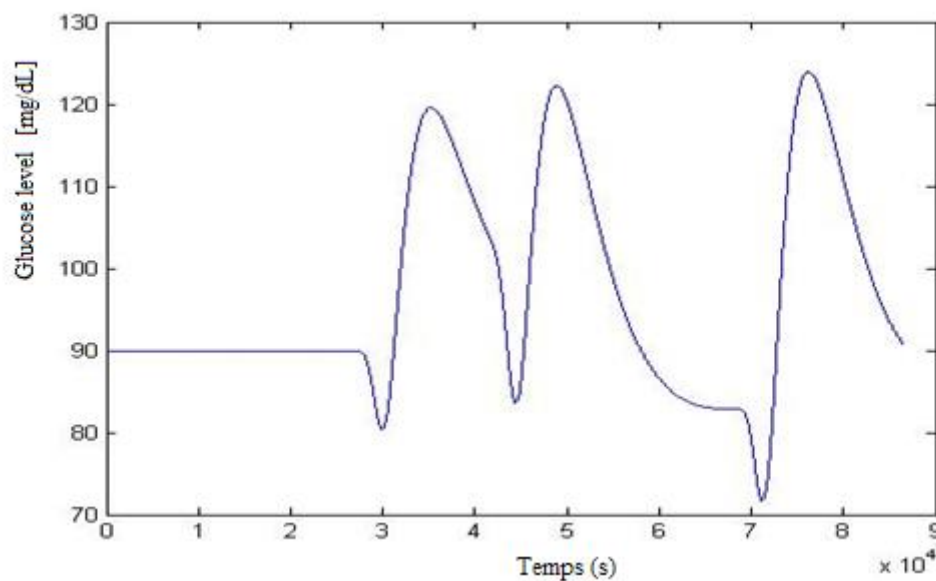


Fig. III. 8: Insulin given before the meal.

III.3.2. Implementation of artificial pancreas

Most of the proposed works of the implementation of this model are done using Matlab/Simulink or java software [72]. It's often due to the complexity of the use of the microcontroller and the huge amount of time needed to implement such methods when incorporating constraints. However, by proposing an optimized algorithm with a reducing size of code and by choosing a low cost microcontroller with low power consumption, one can take the advantages of such miniaturized device. The computed insulin dose required,

according to the insulin dosage computation, is a complex calculation algorithm. Additional purpose is to check if the computed dose is allowed to be administered. The different members of the 8051 microcontroller family are suitable for a huge range of projects [73]. In this part, we have chosen to work with 8051 microcontroller in order to simulate it. We have also simulated this model using Keil development. We have programmed the controller by creating an embedded C program that act as an artificial pancreas. Its main role was to regulate the blood sugar level by insulin injection (figure III.4).

After the sensor reads the level of glucose and sends the result to the controller, the latter calculates the necessary dose to be delivered to maintain an existing trend in blood sugar levels between 70 mg/dl and 110 mg/dl. Using readings from the embedded sensor, the system automatically measures the level of the glucose in the sufferer's body. Consecutive readings are compared in order to provide insulin when needed. Insulin is only delivered in circumstances where it appears that the level of glucose is likely to go outside this range. The dose given as glucagon, starts to be active when the level of glucose is below 60 mg/dl .

The simulation setup is shown in figure III.9. Inputs to the virtual patient were glucose and insulin, whereas the output $G(t)$ was blood glucose concentration. A parallel prediction algorithm was used to calculate the future plasma glucose concentration values sent to the controller. Estimated and measured blood glucose from the simulator were used by the controller to determine the doses of insulin or glucagon to be given to the virtual patient.

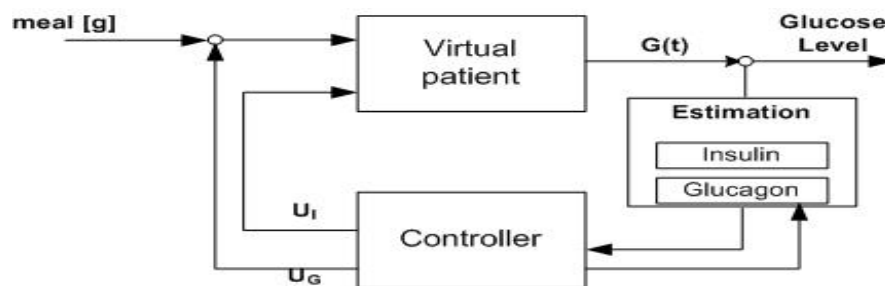


Fig. III. 9: Simulation closed loop.

The Artificial Pancreas Microcontroller (APM) chooses automatically the correct dose to be injected according to glucose level and basing on rules of sugar level of the control algorithm. The APM tests the glucose level many times a day, and according to its algorithm it decides whether to inject or not and if yes, the necessary computed dose to be injected. The table below (table III.9) describes the mainly basic rule that was taken into consideration to build our control algorithm.

Rule #	Description
1	Low sugar level
2	Medium sugar level
3	High sugar level
4	Increasing in sugar level
5	Stable sugar level
6	Falling sugar level
7	Rate of increase is falling
8	Rate of increase is increasing
9	Rate of decrease increasing
10	Rate of decrease decreasing
11	Administer computed dose
12	Manage maximum daily dose

Table III. 9: Control algorithm main rules.

We define the basic rules for the quality of service with the ability of the system to satisfy well-established medical decision criteria. This QoS must respect the value in table III.10 below:

Parameters	Value	Description
<i>SafeMin</i>	70 mg/dL	Safe minimum level of blood sugar
<i>SafeMax</i>	110 mg/dL	Safe maximum level of blood sugar
<i>MaxDailyDose</i>	25 IU	The maximum dose of insulin in 24 h
<i>MaxSingleDose</i>	5 IU	Maximum dose in a single injection
<i>MinDose</i>	1 IU	The min dose to maintain an existing trend in blood sugar
<i>MaxDose</i>	4 IU	The max dose to maintain an existing trend in blood sugar

Table III. 10: Specific parameters.

A part of the result is shown in the graph below (figure III.10a) (with the same scenario for the meal as in the first model). It shows the result after applying the control algorithm with the same scenario for meal as in the first model. The results are of course worse than the

previous case with human decision. We can see that after breakfast and dinner the blood glucose was too high (cf table III.1). Indeed, the human injection has been decided one hour before the meal, the APM could only detect a change in *PV* after the consequence on the blood glucose after having the meal, which means after the glucose level is risen.

Our goal was not to develop an ideal Artificial Pancreas which will have probably to anticipate the meal in order to regulate blood glucose in the good interval. In this Work, the controller was used only in order to test the performance of all the components of the system.

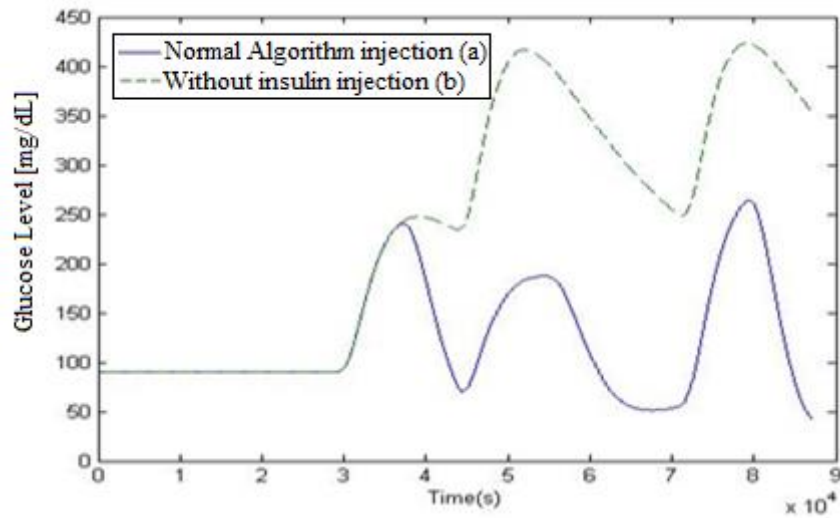


Fig. III. 10: With and without insulin injection.

We can see in figure III.10.b what will happen without the APM, the blood glucose reaches a very high value after lunch and dinner. As previously explained the APM is not perfect but is acceptable as a proof of concept for our tests.

In order to formulate the effective tester, we have changed some parameters. If a problem occurs in an actuator level, these produce an addition of 1 or 2 units or more on the amount of dose injected. Another case is considered, if a problem occurred on the sensor level, the glucose level measured is raised for example 10% or 20% (Figure III.11).

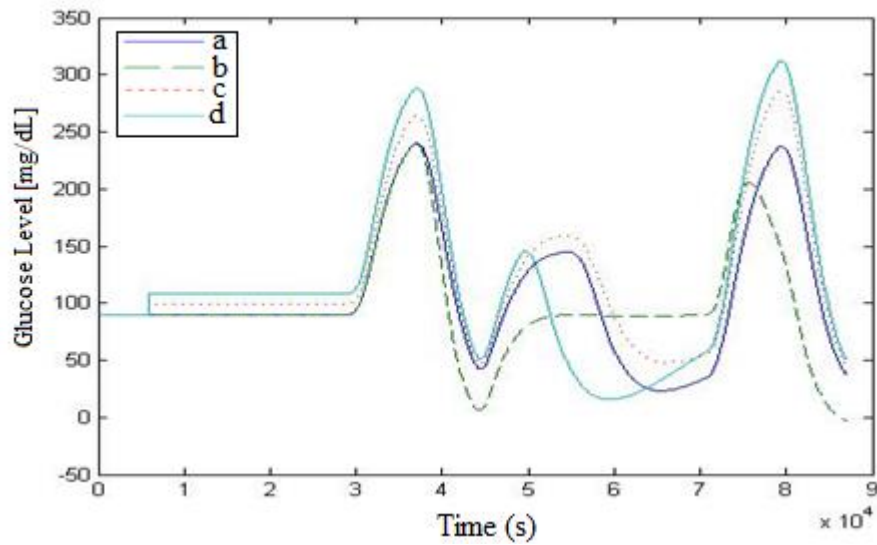


Fig. III. 11: Simulations results: (a) +1 unit injected (b) +2 units injected (c) +10% on glucose level (d) +20% on glucose level.

These show how the changing in injected insulin dose or having a problem in a hardware or software level can affect the system. These results help to test the system and make a good change in performance and test the response of the system in many cases.

III.3.3. Implementation of sensors and actuators

Sensors and actuators are the essential peripherals connected. There are often used in medical technology, for example in micro-pumps, ultrasonic emitters, etc. They are broadly termed transducers and are essential devices that convert one form of energy into another.

The most widespread example of a commercial biosensor [74] is the blood glucose biosensor, which uses an enzyme to breakdown the blood glucose/sugar into its metabolites. The GlucoWatch® G2TM Biographer (GWB) and the Continuous Glucose Monitoring System (CGMS) have both been developed to assist in closer monitoring of glucose levels [75, 76]. Nano-bio sensor ASIC operates at +5V input and provides an output of 0 to 500 mV for absolute sensor output of 0 to 500 nano-amps or pico-amps.

The biosensor can be divided into three components [77, 78]. As shown in Figure III.12, the First component is the biological element (Bio-receptor) which is used to bind the target molecule. It must be highly specific, stable under storage conditions, and immobilized. The molecular recognition elements include receptors, enzymes, antibodies, nucleic acids, microorganisms and lectins. The second components is the physiochemical transducer that acts as an interface, measuring the physical change that occurs with the reaction at the bio-receptor then transforming that energy into a measurable electrical output. The five principal

transducer classes are electrochemical, optical, thermometric, piezoelectric, and magnetic. The Third component is the detector where signals from the transducer are passed to a microprocessor where they are amplified and analyzed. The data is then converted to concentration units and transferred to a display or/and data storage device.

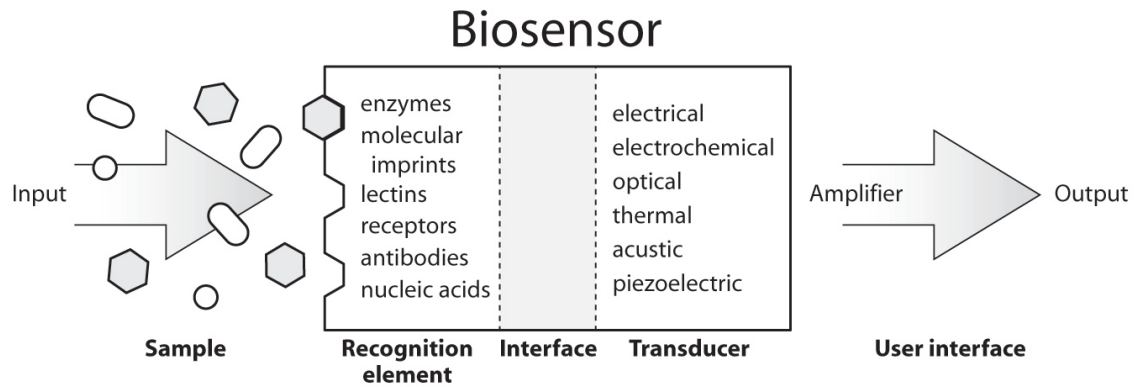


Fig. III. 12: Configuration of a biosensor.

The overall performance of a drug delivery system is obviously dependent on the available actuators. Many different means to transform energy into motion have been explored [79]: Electromagnetic, Electrostatic, thermo-mechanical, Phase Change, Piezoelectric, Shape Memory Alloy (SMA), Magnetostrictive, Electro-rheological, Electro-hydrodynamic, Diamagnetism (Meissner Effect).

For the actuators, we will be interested in electromagnetic motors as they are available in small and compact versions [80]. For example Fritz Faulhaber has set a new standard within the category of DC-Micromotors with graphite commutation. The motor delivers a torque of up to 120 mN.m while measuring just 32 mm in diameter and 72 mm in length.

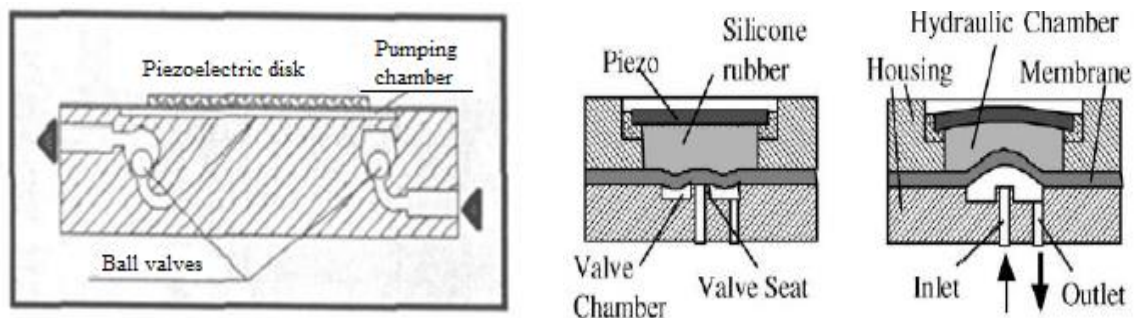


Fig. III. 13: Micro-pump & micro-valve with piezoelectric actuator.

The medical devices such as a micro-pump to extract blood through a tube have a structure which needle and pump part are mutually separated. Bio-actuators can also be implanted into a living body (in vivo) to perform special functions.

An actuator is a device that converts an electrical signal into an action. It can create a force to manipulate itself, other mechanical devices, or the surrounding environment to perform some useful function [81]. In the case of this study, we are interested on pump insulin used for administration of insulin in case of mellitus diabetes. The blood sugar level is usually measured in either mmol/L or mg/dL. We can now find the ratio between the two units going step by step as defined by:

$$\text{mmol} \times 0,18 = \text{g} / \text{l}$$

$$\text{g} / \text{l} \times 5,56 = \text{mmol}$$

Table III.11 shows the range value of glucose-insulin level measured before a meal or two hours after meal, for non-diabetics and diabetic persons. This table shows that the normal value of glucose level for non-diabetics person is 70-110mg/dL and greater than 140 mg/dL for diabetics.

Glucose test	non-diabetics	Diabetic person
before meal	70-110mg/dL	> 140mg/dL
2 hours after meal....	<110mg/dL	> 200mg/dL

Table III. 11: Glucose test before and after meal.

In this model, the systems can be tested in many levels by the implementation of test scenarios based on parametric models of activities, taking meals and the setting of the patient (weight, pathology, etc.).

Sensors and actuators were simulated using mathematical functions based on many conditions states in order to simulate their functions. The purpose of this simulation is to have a complete environment with the ability to co-simulate the glucose insulin model.

For example, table III.12 below calculates the computed dose to be given by comparing the current and previous glucose level. Set Q_1 the current Glucose level, Q_2 the previous Glucose level, and “CompDose” the computed dose to be injected.

Part of the control algorithm function where sugar is ok
<pre> if ($Q_1 > safeMax$) { // If Sugar level increasing if ($Q_1 > Q_2$) { // If dose is rounded to zero, deliver the min dose } } </pre>

```

if ((Q1 - Q2) / 4 == 0)
{
    compDose=minDose;
}
else if ((Q1 - Q2) / 4 > 0)
{
    //Set the amount to deliver
    if ( Q1>110 && (Q1- Q2)>70)
    {
        compDose=MaxDose +1;
    }
    else if ( Q1>SafeMax && (Q1- Q2)>SafeMin/2 && (Q1- reading2)<=SafeMin)
    {
        compDose=MedDose +1;
    }
    else if ( Q1 > SafeMax && (Q1- Q2)<= SafeMin/2)
    {
        compDose= minDose +1;
    }
}
}
}

```

Table III. 12: Function that Calculate the computed dose.

III.4.Programmed Software

When setting the output of result of keil experimentation as a file, the values of glucose are joined together without space but in a specific format. So we need to find a way to take these values that are on most scenarios around 80 935 values in each simulation, and represented as an array. This array will look like this one:

Str = [90.374962 90.381195 90.387421 90.393639]

We have programmed a vb.net program (figure III.14) that helps to take the output of the Keil and plot it in Matlab as a graph. This program reads as input the source file generated by keil as output that contains the glucose level values, and generates a vector of all the values organized in a way that Matlab accept it. We also use a function to calculate the percentile with the Mean, the standard deviation (std), the standard error (stderr) and the median, a part of this method of values calculated is used in the grid representation section described in this thesis.

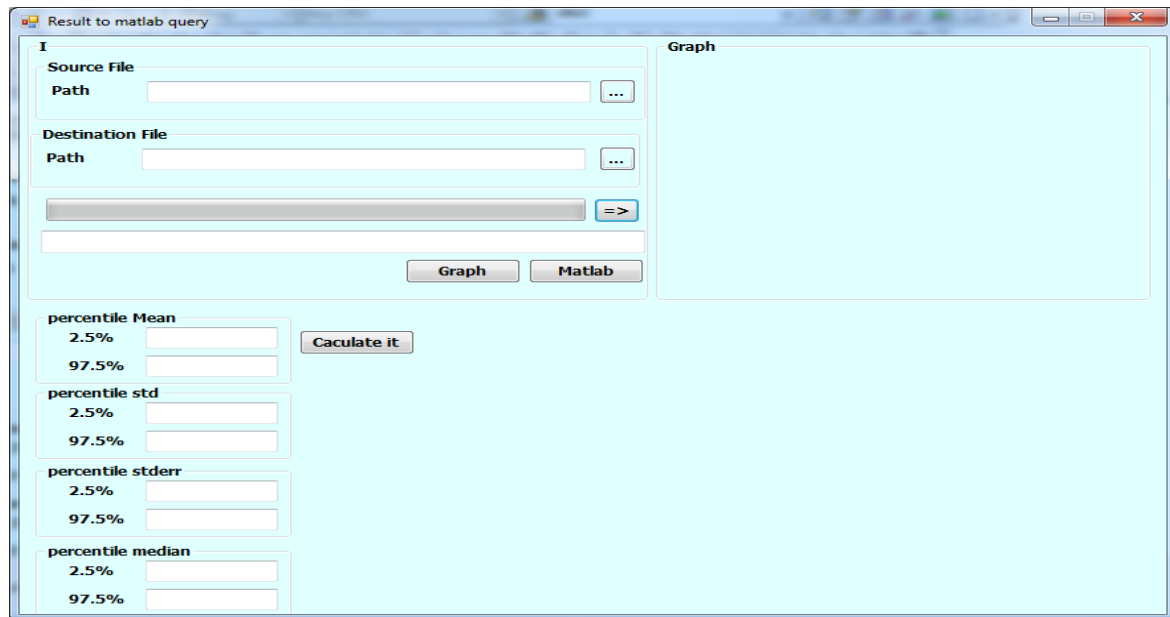


Fig. III. 14: Vb.net program.

III.5.Models and methods of experimentation

Defining a usefulness tools for comparing the use of such sensors, actuators and controllers, can be very useful in testing fields. The model presented in our study helps to simulate the biomedical equipment, by simulation the patient level, sensors, actuators levels, and simulating the electronic hardware.

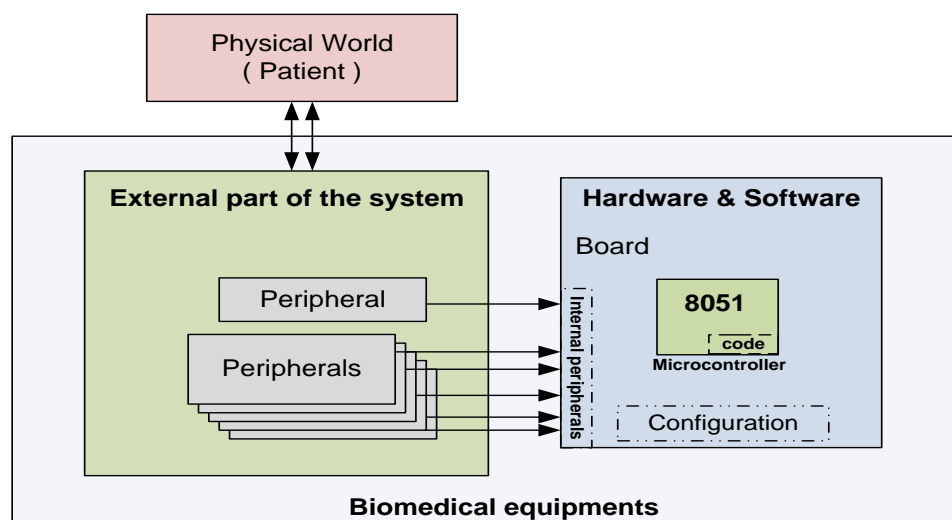


Fig. III. 15: Global view of biomedical equipments.

Figure III.15 represents the complete system that contains three components. The first component described as “Physical world” which represents the patient level. The second

component “External part of the system” where the peripherals can be connected like sensors and actuators, represents the second level. The last component “Hardware & Software” represents the microcontroller with the *C* program code and configuration at this level.

The model of the patient is based on “Hovorka” model. The system (set of differential equation) has been implemented using Keil simulator program and simulated using cycle by cycle keil on architecture type 8051. It can be tested in many levels by the implementation of test scenarios based on parametric models of activities, taking meals and the setting of the patient (weight, pathology, etc.).

We try to implement in-silico technique on a virtual type 1 diabetes mellitus (T1DM) patient. The in-silico experimentation [82, 83, 84] presents many advantages by providing a higher work productivity, minimum cost and more accurate simulations through more sophisticated models.

III.5.1. Experimental conditions

This in-silico study, for the “Hovorka” model, considered as a nonlinear meal simulation. Model parameters were obtained from the model in order to reproduce as faithfully as possible a T1DM patient glucose metabolism. Figure III.16 shows the closed-loop system of this model, simulated sensor and insulin pump and the controller. The Hovorka model plays the role of the patient to be simulated. The simulated sensor reads the data of the patient and sends them to the controller. The controller has a role to controller the system by sending command to the simulated insulin pump.

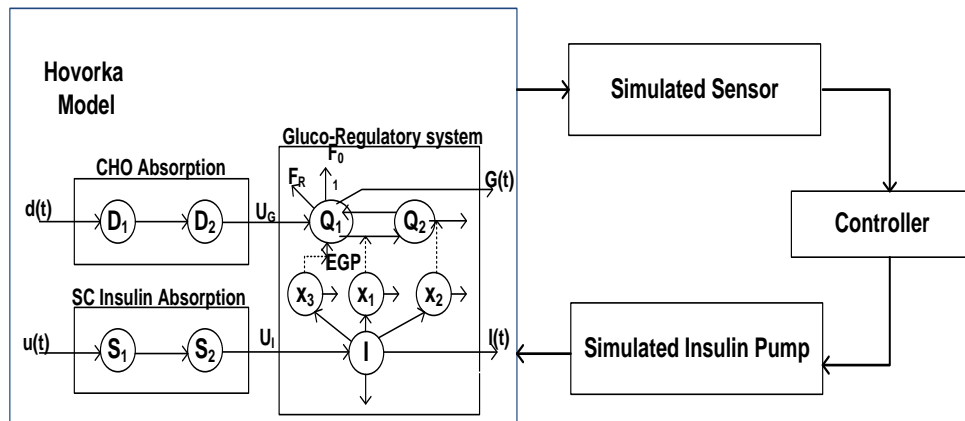


Fig. III. 16: In-silico with Hovorka model.

We assume that the simulation starts at midnight with the virtual patient in a steady specific condition. Table III.13 below represents the virtual patient within 4 days in-silico

visit. The performance of closed-loop glucose control is tested during these days. Every day, a certain amount of carbohydrate is being served, for example in day 2, a big lunch of 100 [g] is consumed. The amount of meal eaten and the insulin unit injected are different from one day to another. By trying, for example, to increment the portions of meals eaten in a day, while maintaining the insulin unit given. On the other hand, an increment the insulin injection and increment or maintain carbohydrate amount. This helps to determine the side effect of them in the state of the patient.

Day	Meal Time	CHO (g)	Injection time	Insulin Unit (IU)	IU/day manual
1	8:00	45	8:00	2	8
	12:00	70	12:00	3	
	19:00	70	19:00	3	
2	8:00	45	8:00	2	8
	12:00	100	12:00	3	
	19:00	70	19:00	3	
3	8:00	45	8:00	2	10
	12:00	100	12:00	5	
	19:00	70	19:00	3	
4	8:00	75	8:00	4	16
	12:00	100	12:00	6	
	19:00	100	19:00	6	

Table III. 13: In-silico table data.

Noting that meals and corresponding insulin doses calculated according to an insulin-to-carbohydrate ratio (ICR) 1:10 were administered complying with the scheme in Table III.13. Figure III.17 shows the simulated data, the glucose level measured during 4 days must be

maintained between the red lines. We have noticed that the curve in some part of this simulation is high or low due to specific parameter entry.

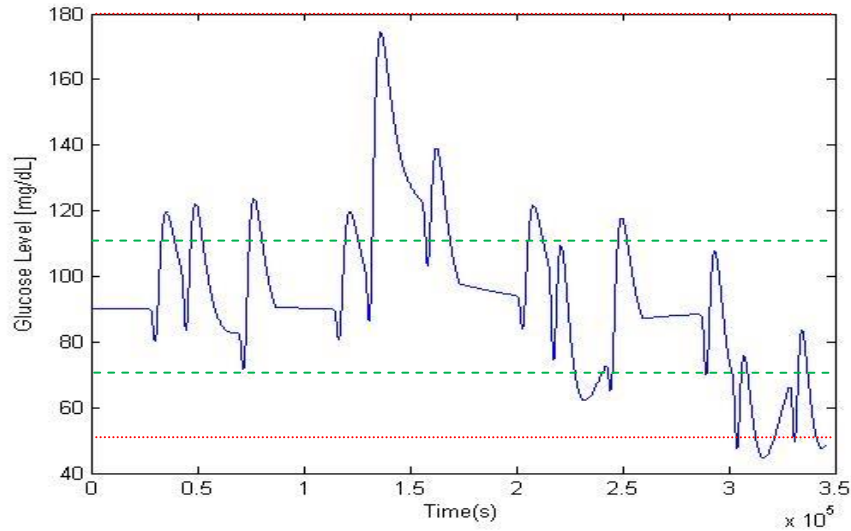


Fig. III. 17: Simulated patient data during 4 days.

The simulation during 4 days is so important to our tester, because the simulation in a day life cannot give a precious answer. It might be during a simulation of a day life, a part of this day for example during breakfast, the glucose level is so high and during dinner is so low, so in total the QoS is in good value.

So to have a better QoS value we try to illustrate our results from 4 days together and for each day during this simulation we compare the results to have more precision and more efficiency.

III.6. Conclusion

In this chapter, we have provided a model that represents a global view of the biomedical equipment, based on repeated simulation to minimize the error of the glucose-insulin model. The results show the feasibility of co-simulation and therefore the ability to validate a finely embedded system without any biomedical risk taking on patients. We implement an environment allowing the co-simulation system of this model, applying in-silico technique using microcontroller system simulator.

We will try to present, a model-based analysis tool that analyze the performance of the biomedical devices and use the resulting models to identify the usage of such sensors, actuators or controllers.

Chapter IV: Simulations results & QoS measurement practices

IV.1. Introduction

Any measure, whether quantitative or a hard fault more subjective judgment of a soft fault, will not be perfect [85, 86, 87]. When making a decision on the conformity of an embedded system with a specification limit, any measurement uncertainty will lead to a risk that incorrect decisions will be made in compliance. This decision risk must be assessed not only in terms of relative consumer provider of respective risks, but also in terms of impact.

Innovations in biomedical technologies are seen as being able to provide solutions to improve the quality and the efficiency of healthcare systems [20].

Where embedded system failure can lead to serious consequences, regulation and strict specifications of product quality are set and compliance based on testing of actual product with these requirements will need to be made both at initial type approval and subsequent verification.

Verification is a major concern for many embedded systems. Hence the importance of validation of these systems, that is to say, if to do a test, verification and certification. So there is a real and pressing need to develop methods and effective tools for the validation of embedded systems. Formal methods are developed for timed systems, but are limited to systems of low complexity.

In brief, the main objective in the domain modeling for embedded systems is to study formal models in order to describe these systems and their constraints (design, specification), to build (programming, simulation, synthesis, implementation), and analyze (validation, verification). When making a decision on compliance of an embedded system specification limit, any uncertainty measure will lead to a risk that incorrect decisions will be made of compliance.

In this chapter, the sections show virtual patients generating part, modeling results as well as the control performances in closed-loop achieved exploiting the in-silico patient. It also shows the simulations results and shows the performance of our tester model declared in chapter II for the case studied in chapter III, the glucose level.

IV.2. Clinical cases generation

Many researchers have studied the modeling of glucose-insulin regulation system in order to better understand it, and the causes of diabetes and development of advanced control algorithm to regulate glucose level concentration and investigation of the pathophysiology of diabetes. These models cannot give a meaningful prediction unless its parameters are accurately determined. Identifying virtual patient parameters is done by fitting the model to the patient data and selecting the necessary values, which give the closest fit to the data.

The Hovorka model used has been selected according to the need of the research. As discussed before, this is a complete model for the glucose-insulin system during a meal and it was developed using glucose tracers.

Eight variables were considered as the inputs data providing to each one of them different values in order to simulate the Hovorka model. The $(Q_1, Q_2, S_1, S_2, I, x_1, x_2, x_3, t)$ help to generate data while solving the Hovorka mathematical model. Indeed, solving mathematical equations inside each parameter helps to have “clinically accurate” results.

A simple modification in each parameters helps to have different patient state. The severity of a patient can be modified by changing parameters values. The objective is to have multiple scenarios in order to use them in the implementation part. We have used the suitability of Partially Observable Markov Decision Processes (POMDP) [88] to formalizing the planning of clinical management (figure IV.1), where α_i is the modification values when generation these virtual patients, taking into consideration the approximation range of each parameter. The modification can be done on one parameter or many parameters at the same time. This minor modification is effective to make different patients. Using POMDP it describes a stochastic control process with partially observable (hidden) states. We start from a virtual patient and after the calculation of normalization functions in order to know which zone the patient point is plotted in, we can specify a modification on parameters so to reach a second virtual patient state.

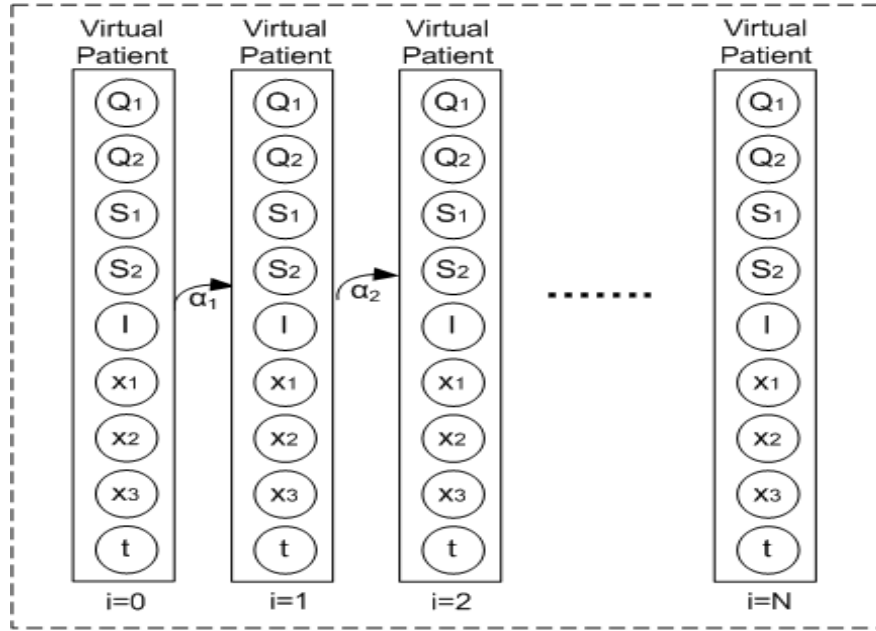


Fig. IV. 1: Virtual patients.

A virtual population of subjects with type 1 diabetes comprises a simulation model of the glucose regulation accompanied by N parameter sets representing N virtual subjects.

The process flow in developing the methods used to generate and simulate glucose-insulin regulatory system is shown in Figure IV.2. Using Keil and a simple program helps to test, debug and simulate it, which can be very useful in testing fields. So the mathematical model is Hovorka using data input vector to have many virtual patients and later an effective tester. To help in generating these patients automatically, we develop a simple program which generates approximately variables for each parameter that compose this model.

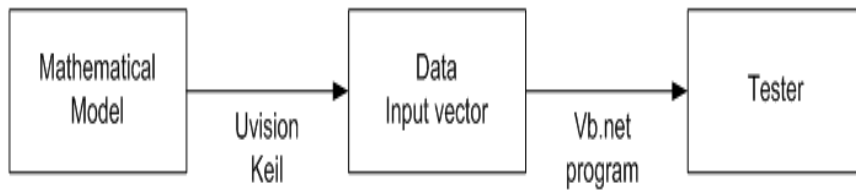


Fig. IV. 2: Outline of the methods used.

Figure IV.3 below shows the set of virtual patients generated using our simple program. Noting that, each element of the vector (Q_1 , Q_2 , S_1 , S_2 , I , x_1 , x_2 , x_3 , t) can be calculated (chapter II Hovorka model) using set of parameters. The F_{01} , K_{12} , K_{a1} , K_{a2} , K_{a3} , K_{b1} , K_{b2} , K_{b3} , K_e are the effective parameters that affect all the system.

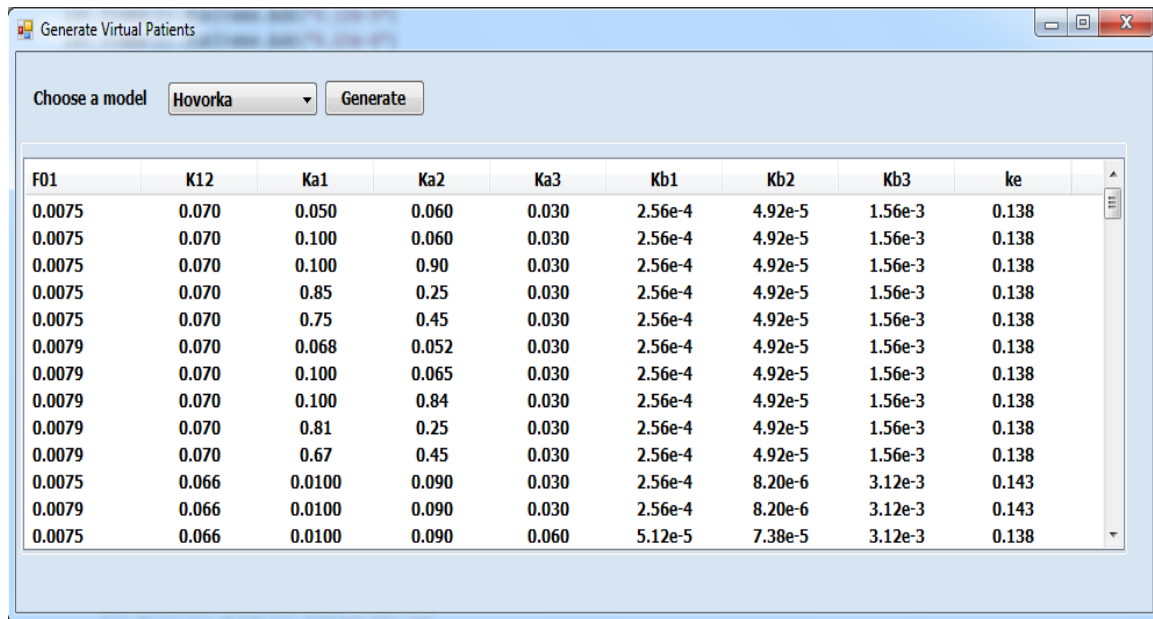


Fig. IV. 3: Software program to generate patients.

The simulation environment consists of a set of virtual patients, a mathematical model and an insulin delivery model or an actuator. The control algorithm interacts with the simulation environment.

IV.2. Virtual patients sample

Table IV.1 shows adjustment parameters of a sample of 6 patients for “Hovorka” model are represented below. Six virtual patients are used in order to analyze the performance of system components and to use them into an in-silico experiment during four days. The data collected is very important in analysis part. There is also a need to improve the quality of the services provided, by ensuring biomedical devices are fit for purpose, which give an opportunity to develop new services or new diagnosis with an objective of upgrading and improvement.

As we notice, the parameters changes are so close to each other which would help to have multiple states of patient.

	1	2	3	4	5	6
F₀₁	0.0075	0.0075	0.0079	0.0079	0.0075	0.0075
K₁₂	0.070	0.066	0.066	0.070	0.070	0.066
K_{a1}	0.050	0.0100	0.0100	0.050	0.080	0.090
K_{a2}	0.060	0.090	0.0100	0.060	0.0100	0.0100
K_{a3}	0.030	0.060	0.060	0.060	0.050	0.060

K_{b1}	2.56e-4	5.12e-5	5.12e-5	2.56e-4	4.096e-4	4.608e-4
K_{b2}	4.92e-5	7.38e-5	8.20e-6	4.92e-5	8.20e-6	8.20e-6
K_{b3}	1.56e-3	3.12e-3	3.12e-3	3.12e-3	2.6e-3	3.12e-3
k_e	0.138	0.138	0.143	0.138	0.143	0.143

Table IV. 1: 6 virtual patients' data.

Each virtual patient can be used as a part of our simulation model, where the system performance of each component will be simulated and tested.

IV.3. Simulation Implementations

In this part, we will try to use this bank of virtual patients and our Vb.net program to calculate percentile values in order to specify each one belongs to which Grid zone (A, Upper B,...) when we apply a day life simulation with normal manual injection. This classification helps to know the state of patient in many cases.

The calculation of total analytical errors, of relative is usually more appropriate for intervals containing high values and calculation of differences is for low intervals. It should identify the 2.5th and 97.5th percentiles used in the calculation of the total error for 95% differences [89, 90]. This 95 % are percentiles, which means that the requirement is for the 95th percentile of the distribution of the differences to be less than the limit stated.

The function `prctile (vect, per)` (table IV.2 that take two parameters the first one is the array values of the simulation and the percentile value as the second parameter for example 2.5.

`prct25p1 = prctile(str1,per)`

Function name	prctile
Function prototype	Func void prctile (str1,per)
Behavior description	Percentile function to calculate the percentile of a simulation
Input	Str1: the simulation array. Per : the percentile value
Output	Percentile value

Table IV. 2: Percentile function definition.

This part of our Vb.net program (figure IV.4) is responsible to calculate the percentile

value, the mean, the standard deviation, the standard error and the median. The purpose of calculating the percentile, and either calculating the standard deviation or using analysis of variance, is to determine the components of imprecision. We can use one of this attributes or the combination of one or two in our function.

The screenshot shows a web-based interface with a light blue background. It contains four distinct sections, each with a title and two input fields. The first section is titled 'percentile Mean' and has input fields for '2.5%' and '97.5%'. To the right of these fields is a button labeled 'Calculate it'. The second section is titled 'percentile std' and also has '2.5%' and '97.5%' input fields. The third section is titled 'percentile stderr' with similar input fields. The fourth section is titled 'percentile median' with '2.5%' and '97.5%' input fields. The interface is clean and uses a simple color scheme.

Fig. IV. 4: Software program to calculate percentile, mean, std, stderr, median.

Part of these functions definitions and the algorithms are described below (codes in Appendix C). Table IV.3 represents the algorithm used to calculate the percentile function.

Algorithm : Calculation algorithm for the percentile function

8. **Let** len be the length of the sorted data array that represent the simulation points and $0 < p \leq 100$ be the desired percentile.
9. **If** $p > 100$ or $p < 0$ **then return** unique array element. **Otherwise**
10. Compute the estimated percentile position / $position = (len + 1) * p / 100$
11. **Set** $n = p / 100 * (len - 1) + 1$;
12. **Let** $left$ be the element in position $\text{floor}(position)$ in the array and **let** $right$ be the next element in the array, where floor is the largest integer less than or equal to the specified position
13. **If** $position \geq 1$ **then** calculate the $left$ and the $right$ value. **Otherwise return** $left$ equal to the *first* element of the vector and $right$ the second.
14. **If** $left$ equal $right$ **then return** the value of $left$. **Otherwise return** $left + (n -$

$$\text{floor}(n) * (\text{right} - \text{left})$$

Table IV. 3: Percentile Algorithm.

Table IV.4 and table IV.5 represents the definition of the function median and its algorithm. Noting that, the median is the middle number of a set of numbers. If there is an even number of entries, it is the average of the two middle numbers.

Function name	Median
Function prototype	Function void Median (NumericArray)
Behavior description	Median function used to calculate the value of the percentile with median
Input	NumericArray: array.
Output	Median value

Table IV. 4: Median function definition.

Algorithm : Calculation algorithm for the median function
<ol style="list-style-type: none"> 1. Let <i>nbcount</i> presents the length of the array, and <i>halfInd</i> be the position index in the half of the vector 2. Let <i>sortarray</i> the array sorted 3. Let <i>pos</i> as an element at a specified index in the sorted array. 4. If the remainder of <i>nbcount</i> divided by 2 equal 0 then <i>median</i> = (pos(<i>halfInd</i>) + pos (<i>halfInd</i> – 1)) / 2. Otherwise 5. Return <i>median</i> = pos(<i>halfInd</i>).

Table IV. 5: Median algorithm.

Table IV.6 and table IV.7 represents the standard deviation function definition and its algorithm. This function is used to calculate the value of the percentile with a standard deviation. It takes as input the simulation data array and gives as output the standard deviation value.

Function name	getStandardDeviation
Function prototype	Function void getStandardDeviation (doubleList)

Behavior description	Standarddeviation function used to calculate the value of the percentile with standard deviation
Input	doubleList: array.
Output	Standard deviation value

Table IV. 6: Standard deviation function definition.

Algorithm : Calculation algorithm for the standard deviation function
<ol style="list-style-type: none"> 1. Let <i>avg</i> be the average value of the array and <i>n</i> the length of the array 2. Let <i>sumderivation</i> as the sum of the derivation of the array 3. For each <i>value</i> in the array do 4. <i>sumderivation</i> \leftarrow <i>sumderivation</i> + (<i>value</i> * <i>value</i>) 5. End For 6. Set <i>sqrt</i> as the square root of a specified number. 7. Return <i>sqrt</i> (<i>sumderivation</i>/<i>n</i> – (<i>avg</i> * <i>avg</i>))

Table IV. 7: Standard deviation function implementation.

In the following two sections, we try to simulate the state of two virtual patients in one day. During this simulation, we simulate the breakfast, lunch and dinner for each one of them, and we compare the value of each meal with the value of the day. The state of a patient may differ during the meals.

IV.3.1. Virtual patient one

In Figure IV.5, we try to illustrate the state of a normal virtual patient. This patient has a glucose level between normal ranges. So if we try to plot this virtual patient during this day, by plotting the day and each meal taken.

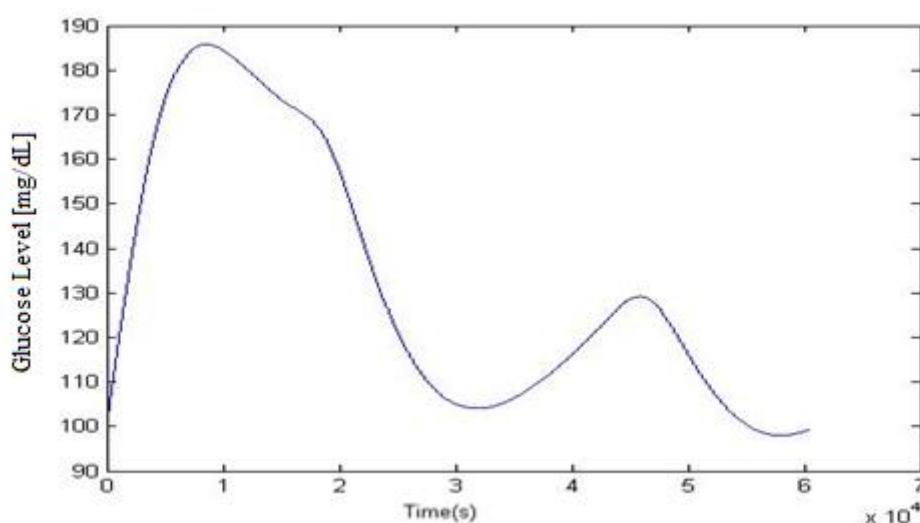


Fig. IV. 5: Virtual patient one.

In figure IV.6, we notice how the state of the patient has been changed during a day in the morning during breakfast, or in the afternoon during lunch or in the evening during dinner. But all the four points are considered in the same zone, which leads to the conclusion that the state of the patient during the day is normal.

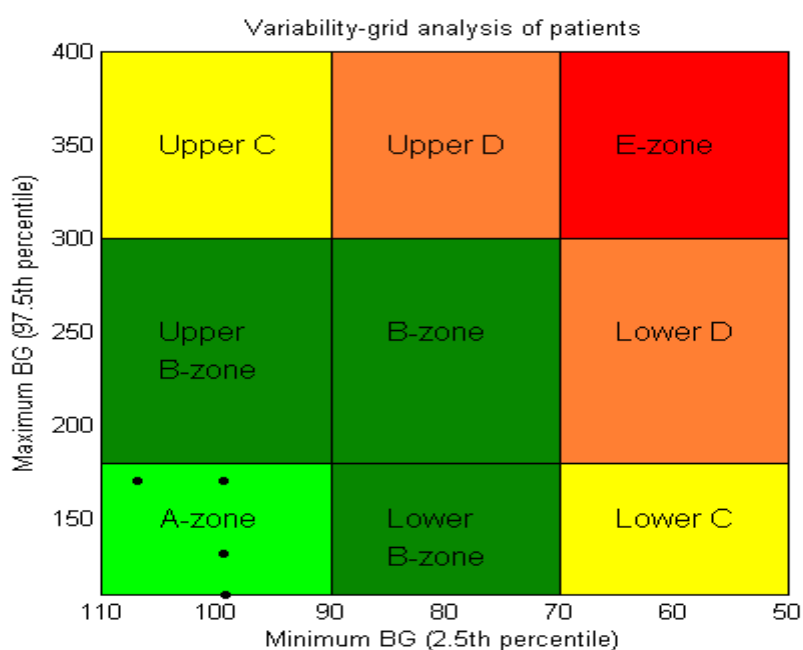


Fig. IV. 6: Normal Virtual patient (1).

IV.3.2. Virtual patient two

In Figure IV.7, we try to illustrate the state of a virtual patient. This patient has a different

glucose level during his day. So if we try to plot this virtual patient during this day too, by plotting all the day and each meal taken.

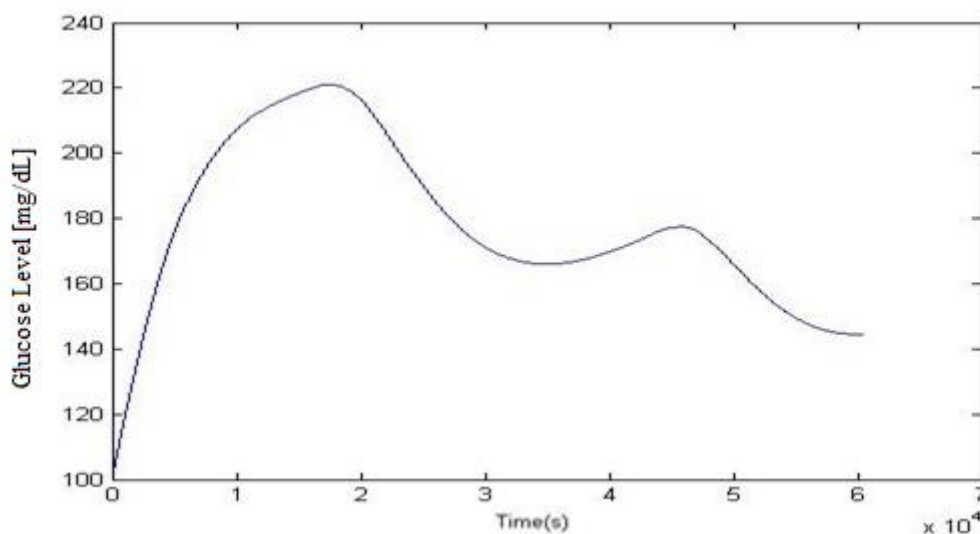


Fig. IV. 7: Virtual patient two.

Figure IV.8 shows the state of the patient during the morning has been in different zone, but all the day the simulation was in Zone A. All of the four points are not in the same zone, which leads us to conclude that the state of the patient during all day has been changed.

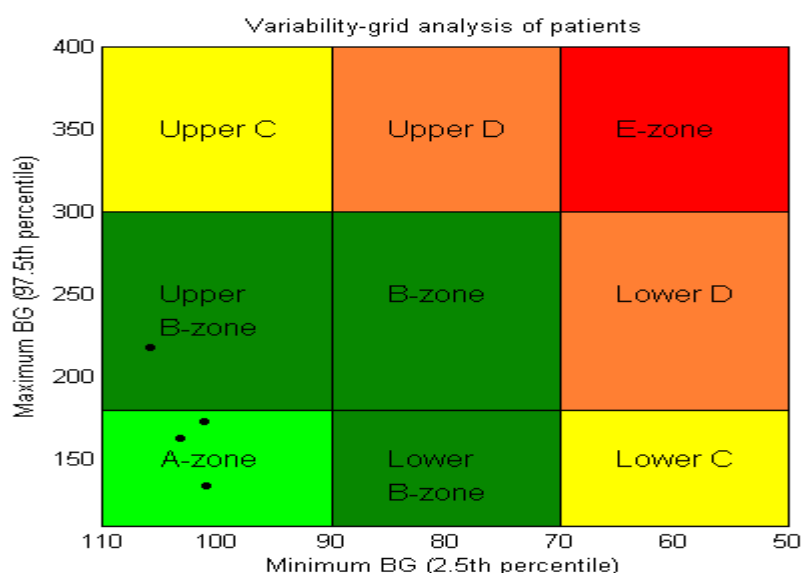


Fig. IV. 8: Normal Virtual patient (2).

IV.4. QoS evaluation implementation

The QoS indicator has an objective, which is the comparison of the diabetic systems with

the normal system. We try to apply the tester model in this system, where the glucose level is the PV controlled and detected.

IV.4.1. Graphic representation

Consider $f(t)$ the function representing the glucose level (blood sugar level) on a time interval $[t_i, t_f]$, already defined in chapter II in paragraph tester model.

The parameters α_m and β_m are predefined constants:

$$\alpha_m = \{600, 400, 360, 300, 270, 200, 180, 150\}$$

$$\beta_m = \{75, 55, 35\}.$$

Set T_{Hyper} the duration in $[t_i, t_f]$ in the hyperglycemia case and T_{Hypo} the duration in $[t_i, t_f]$ in the hypoglycemia case.

Table IV.8 shows IG_H (g) which represents the severity of the patient's case. We implement it in some results done before which would help to formulate an indicator that will be a good step to build the tester. This tester is the first step in building a more comprehensive tester; it will also open a gate to develop a language or equivalent system to facilitate testing of biomedical devices.

The values in table IV.8 showed many severe cases when compared them to normal injection and it will be considered as a reference to graphical representation of the data of a person via glucose and risk traces and plots, and at a group level via Control Variability Grid Analysis. These help to facilitate the extraction of information, and the interpretation of complex and voluminous CGM time series

Graph State IG_H (g)	Normal Algorithm Injection	Without insulin injection	+1 to injected insulin dose	+2 to injected insulin dose	+10% to measured glucose level	+20% to measured glucose level
$\alpha = 150$	1,381,200	9,476,600	895,690	632,250	1,224,500	1,073,600
$\alpha = 180$	682,470	7,849,200	469,810	285,640	615,380	619,350
$\alpha = 200$	398,620	6,782,200	245,530	124,940	367,490	372,140
$\alpha = 270$	-	3,501,700	-	-	44.1	4.8
$\alpha = 300$	-	2,480,600	-	-	-	49.5341

$\alpha = 360$	-	848,700	-	-	-	-
$\alpha = 400$	-	155,490	-	-	-	-
$\alpha = 600$	-	-	-	-	-	-
$\beta = 35$	-	-	35,124	41,413	-	97,599
$\beta = 55$	7,145	-	613,850	111,190	266,360	675,650
$\beta = 75$	484,470	-	949,010	248,310	609,530	902,820

Table IV. 8: IG_H (g) with α , β predefined constant values.

IV.4.2. Grid representation

We analyze each simulation in the purpose of having a graphical representation of the risk assessment for the patient due to choices on sensors, actuators. We use this grid with the percentile function to represent our implementation. The minimum BG is set at the 2.5th percentile, and the maximum BG is set at the 97.5th percentile of the BG distribution. We determine the minimum and maximum inverted percentiles for each simulation, where for each person a point is plotted with x-coordinate the minimum BG and y-coordinate the maximum BG for an observation period. The duration of a simulation is 24 hours; and in a day study each person would get a data point on the grid. We implement our controller in order to analyze its performance.

The World Health Organization (WHO) uses the percentile in their testing and measurement of health data [91]. Percentiles are position measures that are used primarily in educational and health-related fields to indicate the position of an individual in a group. It has been established to be considered as an important indicator for obtaining normalized values.

To simulate the 6 virtual patients from table IV.1 we use the grid analysis and we use Matlab to write the function of simulation. Table IV.9 shows the simulation process of a point in the grid, a point represents one full simulation. Noting that, points exceeding the limits of the plot are plotted on the outer border.

Function name	Clarke4
Function prototype	Function void clarke4 ()
Behavior description	Plotting simulation in a grid using Matlab program
Input	None

Output	None
---------------	------

Table IV. 9: Clarke4 function definition.

Table IV.10 represents the algorithm for error grid implementation using Matlab. Matlab provides built-in functions which help to easy construct the grid (plot, fill, text, set, xlabel, ylabel). We have 9 zones in this grid (Upper C', 'Upper Bzone', 'A-zone', 'Upper D', 'B zone', 'Lower B-zone', 'E-zone', 'Lower D', 'Lower C') and the grid has limit values that the plotted point must respect (more details in Appendix C).

Algorithm: Error grid implementation algorithm in Matlab
<ol style="list-style-type: none"> 1. Let <i>prct25p1</i> as the percentile value for the array for a percentile 2.5 2. Let <i>prct975p1</i> as the percentile value for the array for a percentile 97.5 3. if <i>prct25p1</i><50 then <i>prct25p1</i>=50 end if 4. if <i>prct975p1</i><110 then <i>prct975p1</i>=110 end if 5. Draw label axe x and axe y 6. Use to fill function to draw each zone 7. Draw each zone 'Upper C', 'Upper Bzone', 'A-zone', 'Upper D', 'B zone', 'Lower B-zone', 'E-zone', 'Lower D', 'Lower C', in order to form the grid. 8. Use plot function to <i>plot</i> the two values as a point in the grid <i>plot(prct25p1,prct975p1)</i>

Table IV. 10: Grid function algorithm.

In Figure IV.5, the plot of each simulation as a point in a grid and then splits it into zones defined by their x- and y-coordinate ranges, helps to construct a new error grid and use it to evaluate the accuracy of value measured. In the first implementation, we implement to test the controller in order to analyze its performance.

Noting that, the virtual patient used for the simulations does not include the effect of factors that influence the blood glucose concentration for example like stress, or that person doing exercise.

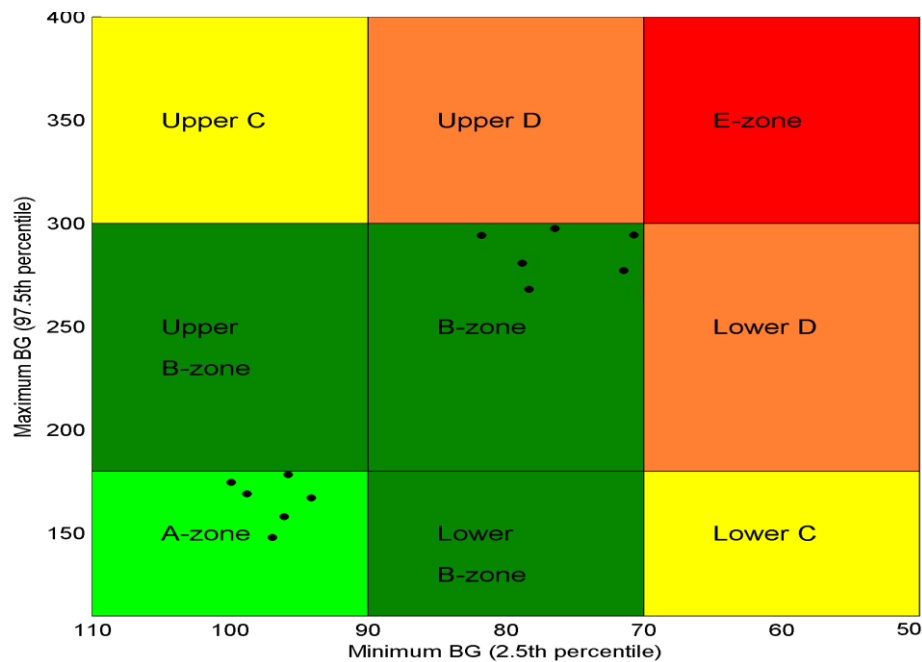


Fig. IV. 9: Graphic representation.

We notice, it's not considered as bad as an automatic injection system but it is not to have an ideal controller, but to arrive to simulate the performance of equipment. The main idea is to consider critical case on hypoglycemia and hyperglycemia, critical sensors, actuators, and controllers.

To be more precise and to have a best representation, we consider that each simulation is not considered only one point in the grid, but it's constructed from one main point and 3 other points that is the representation state of the patient during breakfast, lunch and dinner. We compare then these values in order to see if the three points are in the same Zone as the main point, which mean the main point is considered the desired value. On the other hand, if one of the three points is out of the main point zone, which means that the state of the patient has been changed during the day and more study must be done to monitor and reveal its state.

IV.5. Conclusion

Simulation results show the performance of the tester to validate glucose insulin system in order to avoid risk taking on patients. From these results the tester that we are creating is a robust tool that will improve testing in biomedical fields without having any risk on patients. The plot of each simulation as a point in a grid and then splits it into zones defined by their x- and y- coordinate ranges, helps to construct a new error grid and use it to evaluate the accuracy of glucose level measurements made by patients.

Chapter V: Conclusion

V.1. Contribution

We have declared a new comprehensive methodology for modeling and simulating the human body and medical systems, to have a better understanding of the best way to model and simulate these systems and to detect the performance and the quality of service of all system components. We have also implemented of an environment for co-simulation of a model of the human body, using Hovorka model.

We have created a programmed model that acts as an artificial pancreas. The goal is to have a complete example of a biomedical system in link with the glucose-insulin in order to be able to test testing strategies. In addition to this, we have defined simple indicators for the system-level testing. The purpose of these indicators is to have a reference to the graphical representation of a person's data in connection with the glucose.

In the implementation part, we implement these models in a comprehensive approach to associate the choices made on the biomedical system of indicators of "well-being" associated with patients. We also simulate these models using in-silico experimentation that has many advantages in providing greater labor productivity, as well as the minimum cost and more accurate simulation due to sophisticated models. The simulation of virtual patients with type I diabetes is done during 4 days, receiving breakfast, lunch and dinner every day.

By generating virtual patients, we had generalized the passage of the clinical trial in order to have several scenarios for analyzing performance. We have also created a simple tool to generate many patients taking into consideration the approximation range of each parameter.

We have used the suitability of partially observable Markov decision processes to formalize the planning of clinical management. A virtual population of subjects with type 1 diabetes comprises a simulation model of the glucose regulation accompanied by N parameter sets representing N virtual subjects.

A choice of a more sophisticated tester, by introducing different sensors and actuators scenarios appears as a promising research direction, although the extension of the algorithm to

industrial applications with microcontroller having limited computational capacity requires non-trivial investigation effort.

V.2. Future work

Besides the contributions presented, several open improvements should be undertaken.

The first is aimed to the development of a software application used as a library to test embedded device. This library will contain the main global function used to test and simulate performance of these devices.

The second is related to the adjusted models using data from real patients. In this thesis, an in silico simulation was performed using virtual patient profiles, while another simulation must be performed using the profiles of real patients.

The third is oriented toward the clinical, to improve the assessment correctness of the estimation techniques.

Finally, the use of some other simulations solutions as VHDL-AMS could be a study for future development in order to improve results accuracy.

V.3. Publications

During the research work leading to this thesis, the following conference and journal papers have been published:

✓ Journal papers (1)

- **Ch. El-Gemayel**, F.Jumel, J.Constantin, D.Zaouk, N.Abouchi, “A prototype to generalize clinical cases of the glucose-insulin system”, Biomedical Engineering journal in the Walter de Gruyter (WdG), Volume 59 (s1), 2014.

✓ International conferences (3)

- **Ch. El-Gemayel**, F. Jumel, J. Constantin, D. Zaouk, N. Abouchi, “A global methodology for modeling and simulating medical systems”, 16th International Conference on E-health Networking, Application & Services, **IEEE**, pages 402-407, 2014.
- **Ch. El-Gemayel**, F. Jumel, J.Constantin, A. Tabet, D. Zaouk, N. Abouchi “A new framework for analyzing the performance of the Glucose-Insulin System”, 2nd International Conference, Advances in Biomedical Engineering (ICABME) , **IEEE**, pages 159-162, 2013.
- **Ch. El-Gemayel**, F. Jumel, J. Constantin, A. Tabet, D. Zaouk, N. Abouchi “An in-silico study for glucose-insulin system based on microcontroller using system simulator”, 25th international conference Microelectronics (ICM), **IEEE**, pages 1-4, 2013.

✓ International conferences without published proceedings (1)

- **Ch. El-Gemayel**, F. Jumel, J. Constantin, Y. Zaatar, D. Zaouk, N. Abouchi, “A global architecture for patient simulator systems”, 20th LAAS International science conference, **Oral presentation**, pages 23-24, Hadath-Lebanon, 2014

- ✓ National Conference (1)
 - **Ch. El-Gemayel**, F. Jumel, J. Constantin, D. Zaouk, N. Abouchi, “Co-simulation of Physiological Glucose-Insulin human system and embedded system models including hardware and software components”, GDR SoC-SiP, **Poster** ,Lyon-France, 2013

Bibliography

- [1] L. Bertalanffy. General System Theory. Penguin University Books, Middlesex, England, 1968.
- [2] KITANO Hiroaki. Computational systems biology. *Nature*, 2002, 420(6912), pp. 206-210.
- [3] KITANO Hiroaki. Systems biology: a brief overview. *Science*, 2002, 295(5560), pp. 1662-1664.
- [4] SAUER U., HEINEMANN M., and ZAMBONI N. Genetics. Getting closer to the whole picture. *Science*, 2007, 316(5824), pp. 550-551.
- [5] End of the interlude? *Nature Biotechnology*, 2004, 22(10), pp.1191-1327.
- [6] DUBITZKY Wemer. Understanding the computational methodologies of systems biology. *Briefings in Bioinformatics*, 2006, 7(4), pp. 315-317.
- [7] SUMNER Thomas. *Sensitivity analysis in systems biology modeling and its application to a multi-scale model of blood glucose homeostasis*. Thesis in philosophy, University College London, 2010.
- [8] KREKE Jennifer E., SCHAEFER Andrew J., and ROBERTS Mark S. Simulation and critical care modeling. *Current Opinion in Critical Care*, 2004, 10, pp. 395–398.
- [9] ROSEN Kathleen R. The history of medical simulation. *Journal of Critical Care*, 2008, 23(2), pp. 157–166.
- [10] American Diabetes Association. Standards of medical care in diabetes. *Diabetes Care*, 2010, 33:(Suppl. 1), pp. S11-S61.
- [11] IBRAHIM Ahmmmed Saadi, AZMI Ahmad Arif, DZULKIFLI Muhamad, and BIN ABD RAHMAN Nazrin Naim. Advanced mathematical model of glucose-insulin concentrations in type 1 diabetes using artificial neural networks. *Wulfenia Journal*, 2013, 20(5), pp. 479- 495.
- [12] FARMER Terry G. Jr., EDGAR Thomas F., and PEPPAS Nicholas A. Effectiveness of intravenous infusion algorithms for glucose control in diabetic patients using different

simulation models. *Industrial & Engineering Chemistry Research*, 2009, 48(9), pp. 4402–4414.

[13] SANDHYA, and KUMAR Deepak. Mathematical model for glucose-insulin regulatory system of diabetes mellitus. *Advances in Applied Mathematical Biociences*, 2011, 2(1), pp. 39-46.

[14] PASSIMENT M., SACKS H., and HUANG G. Medical simulation in medical education: results of an AAMC survey. *Association of American Medical Colleges*, 2011, pp. 1-48.

[15] WILINSKA Malgorzata E., CHASSIN Ludovic J., ACERINI Carlo L., ALLEN Janet M., DUNGER David B., and HOVORKA Roman. Simulation environment to evaluate closed-loop insulin delivery systems in type 1 diabetes. *Journal of Diabetes Science and Technology*, 2010, 4(1), pp.132-144.

[16] RUSSELL Steven J., EL-KHATIB Firas H., NATHAN David M., MAGYAR Kendra L., JIANG John, and DAMIANO Edward R. Blood glucose control in type I diabetes with a bihormonal bionic endocrine pancreas. *Diabetes Care*, 2012, 35(11), pp. 2148-2155.

[17] DALLA MAN Chiara, RAIMONDO Davide M., RIZZA Robert A., and COBELLI Claudio. GIM, simulation software of meal glucose insulin model. *Journal of Diabetes Science and Technology*, 2007, 1(3), pp. 323-330.

[18] GOLDSMAN David, and TOKOL Gamze. Output analysis: output analysis procedures for computer simulations. In *Proceedings of the 32nd Conference on Winter Simulation*, Orlando, FL, 10-13 December 2000, pp. 39-45.

[19] *Standards Analysis: Biomedical Technologies Sector, Luxembourg*. ILNAS/ANEC, Version 2.0, October 2013, 162 pages.

[20] THOMSON Sarah, OSBORN Robin, SQUIRES David, and JUN Miraya. *International Profiles of Health Care Systems*. The Commonwealth Fund. November 2013, 138 pages.

[21] PATEL Jasbir N., KAMINSKA Bozena, GRAY Bonnie, and GATES Byron D. Electro-enzymatic glucose sensor using hybrid polymer fabrication process. In *Proceedings of the 14th IEEE International Conference on Electronics, Circuits and Systems*, Marrakech,

Morocco, 11-14 December 2007, pp. 403-406.

[22] CHANG-YEN David A., and GALE Bruce K. Integrated optical glucose sensor fabricated using PDMS waveguides on a PDMS substrate. In *Proceedings of SPIE, Vol 5345, Microfluidics, BioMEMS, and Medical Microsystems II*, San Jose, CA, 2003, doi: 10.1117/12.530062

[23] CHIA Chee W., and SAUDEK Christopher D. Glucose sensors: toward closed loop insulin delivery. *Endocrinology Metabolism Clinics of North America*, 2004, 33(1), pp. 175–195.

[24] ROBBINS W.P. High-displacement piezoelectric actuator utilizing a meander-line geometry II. Theory. *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, 1991, 38(5), pp. 461-467.

[25] THIELICKE Ernst, and OBERMEIER Ernst. Microactuators and their technologies. *Mechatronics*, 2000, 10(4-5), pp. 431-455.

[26] HARIKUMAR R., SUDHAMAN V.K., GANESH BABU C. FPGA synthesis of fuzzy (PD and PID) controller for insulin pumps in diabetes using cadence. *International Journal of Soft Computing and Engineering*, 2012, 1(6), pp. 324-331.

[27] VERHOEF Marcel. *Modeling and Validating Distributed Embedded Real-Time Control Systems*. 184 pages. Thesis in Computer Science, Institute for Programming Research and Algorithmics (IPA), 2009.

[28] STHAMER Harmen, BARESEL André, and WEGENER Joachim. Evolutionary testing of embedded systems. *14th Annual International Internet & Software Quality Week*, San Francisco, CA, 29 May - 1 June 2001.

[29] YU Tingting. *Testing Embedded System Applications*. 185 pages. Thesis in Computer Science, University of Nebraska, 2010.

[30] CHENG Michael. *Medical Device Regulations: Global Overview and Guiding Principles*. Geneva, World Health Organization, 2003.

[31] HESTER Robert L., ILIESCU Radu, SUMMERS Richard, and COLEMAN Thomas G. Systems biology and integrative physiological modeling. *The Journal of Physiology*, 2011,

589(5), pp. 1053-1060.

[32] OTTESEN Johnny T., OLUFSEN Mette S., and LARSEN Jesper K. *Applied Mathematical Models in Human Physiology*. BioMath-Group, Department of Mathematics and Physics, Roskilde University, Denmark, October 3, 2006, 292 pages

[33] KOHL P., CRAMPIN E.J., QUINN T.A., and NOBLE D. Systems biology: an approach. *Clinical Pharmacology Therapeutics*, 2010, 88, pp. 25-33.

[34] TORTORA Gerard J., and DERRICKSON Bryan H. *Principles of Anatomy and Physiology*. 12th edition. John Wiley & sons, Inc, 2008, 744 pages.

[35] BORON Walter F., and BOULPAEP Emile L. *Medical Physiology*. 2nd edition, Elsevier Health Sciences, 2012, 1352 pages.

[36] DANGELMAYR Gerhard, and KIRBY Michael. *Mathematical modeling: A Comprehensive Introduction*. Prentice Hall, 2003, 161 pages.

[37] MAZUR James E. Mathematical models and the experimental analysis of behavior. *Journal of the Experimental Analysis of Behavior*, 2006, 85(2), pp. 275-291.

[38] PALUMBO Pasquale, DITLEVSEN Susanne, BERTUZZI Alessandro, and DE GAETANO Andrea. Mathematical modeling of the glucose–insulin system: A review. *Mathematical Biosciences*, 2013, 244(2), pp. 69–81.

[39] Keil Elektronik GmbH and Keil Software. *Getting Started and Create Applications*. 1999, 172 pages.

[40] Website <http://www.keil.com>, last accessed, March 2014.

[41] KRATZIG Gregory p. and HUDY Christine. From Theory to Practice: Simulation Technology as a Training Tool in Law Enforcement. *Police Organization and Training*. Springer, chapter 5, 2012, 214 pages

[42] LAW Averill M. *Simulation Modeling and Analysis*. 4th edition, McGraw-Hill, 2007, 768 pages.

[43] ENGELBRECHT Gerhard, BISBAL Jesus, BENKNERY Siegfried, and FRANGI

Alejandro F. Towards negotiable SLA-based QoS support for biomedical data services. In *Proceedings of the 11th IEEE/ACM International Conference on Grid Computing*, Brussels, Belgium, 25-28 October 2010, pp. 259-265.

[44] O'DONOGHUE John, and HERBERT John. A QoS data management system within a pervasive medical environment. *Advances in Pervasive Computing 2006, Adjunct Proceedings of Pervasive 2006*, Dublin, Germany, 7-10 May 2006, 207, pp. 193-198.

[45] Liang, X.; Balasingham, I. A QoS-aware Routing Service Framework for Biomedical Sensor Networks. In *Proceedings of the IEEE International Symposium on Wireless Communication Systems (ISWCS)*, Trondheim, Norway, 17–19 October 2007, pp. 342–345.

[46] McLAUGHLIN Curtis P., and KALUZNY Arnold D. *Continuous Quality Improvement in Health Care: Theory, Implementation, and Applications*. 2nd edition, Jones & Bartlett Learning, 2004, 659 pages.

[47] SCHMITT Jens B., GOLLAN Nicos, and MARTINOVIC Ivan. A new service curve model to deal with non-FIFO systems. Springer, 2009, pp. 143-154.

[48] BEKKI Jennifer McNeill, MACKULAK Gerald T., and FOWLER John W. Indirect cycle-time quantile estimation for non-FIFO dispatching policies. In *Proceedings of the Winter Simulation Conference*, Monterey, CA, 3-6 December 2006, pp. 1829-1835.

[49] Clarke WL, Cox D, Gonder-Frederick LA, Carter W, Pohl SL: Evaluating clinical accuracy of systems for self-monitoring of blood glucose. *Diabetes Care* 10:622–628, 1987.

[50] PARKES J.L., SLATIN S.L., PARDO S., and GINSBERG B.H. A new consensus error grid to evaluate the clinical significance of inaccuracies in the measurement of blood glucose. *Diabetes Care*, 2000, 23(8), pp. 1143-1148.

[51] KOVATCHEV Boris P., GONDER-FREDERICK Linda A., COX Daniel J., and CLARKE William L. Evaluating the accuracy of continuous glucose-monitoring sensors: continuous glucose–error grid analysis illustrated by TheraSense Freestyle Navigator data. *Diabetes Care*, 2004, 27(8), pp. 1922-1928.

[52] MAGNI Lalo, RAIMONDO Davide M., DALLA MAN Chiara, BRETON Marc, PATEK Stephen, DE NICOLAO Giuseppe, COBELLI Claudio, and KOVATCHEV Boris PP. Evaluating the efficacy of closed-loop glucose regulation via control-variability grid

analysis. *Journal of Diabetes Science and Technology*, 2008, 2(4), pp. 630-635.

[53] KLONOFF David C. The need for clinical accuracy guidelines for blood glucose monitors. *Journal of Diabetes Science and Technology*, 2012, 6(1), pp.1-4.

[54] ABBES Amira Kheriji, BOUANI Faouzi, and KSOURI Mekki. A microcontroller implementation of constrained model predictive control. *World Academy of Science, Engineering and Technology*, 2011, 5(8), pp. 655-662.

[55] Pound Pandora, Ebrahim Shah, Sandercock Peter, Bracken Michael B, Roberts Ian. Where is the evidence that animal research benefits humans?, *British Medical Journal*. 2004, 328(7438), pp. 514-517.

[56] Website <http://www.mismr.org/educational/biomedres.html>, last accessed, May 2014

[57] COBELLI Claudio, DALLA MAN Chiara, SPARACINO Giovanni, MAGNI Lalo, DE NICOLAO Giuseppe, and KOVATCHEV Boris P. Diabetes: Models, Signals, and Control. *IEEE Reviews in Biomedical Engineering*, 2009, 2, pp. 54–96.

[58] MAGNI Lalo, RAIMONDO Davide M., BOSSI Luca, DALLA MAN Chiara, DE NICOLAO Giuseppe, KOVATCHEV Boris, and COBELLI Claudio. Model predictive control of type 1 diabetes: an in silico trial. *Journal of Diabetes Science and Technology*, 2007, 1(6), pp. 804-812.

[59] Chiara Dalla Man, Davide M. Raimondo, Robert A. Rizza, Claudio Cobelli, “GIM, Simulation Software of Meal Glucose–Insulin Model”, *Journal* May 2007

[60] OKUDA Yasuharu and QUINONES Joshua. The use of simulation in the education of emergency care providers for cardiac emergencies, *International Journal of Emergency Medicine*, 2008, 1(2), pp. 73-77.

[61] HAAS Linda, MARYNIUK Melinda, Beck Joni... National Standards for Diabetes Self-Management Education and Support, *Diabetes Care*, 2014, 37(Suppl. 1), pp. S144-S153.

[62] RAMCHANDANI Neesha and HEPTULLA Rubina A. New technologies for diabetes: a review of the present and the future. *International Journal of Pediatric Endocrinology*, 2012, 2012(1):28. doi: 10.1186/1687-9856-2012-28.

- [63] STEIL Garry, REBRIN Kerstin, and MASTROTOTARO John J. Metabolic modelling and the closed-loop insulin delivery problem. *Diabetes Research and Clinical Practice*, 2006, 74, pp. S183-S186.
- [64] ELLERI Daniela, DUNGER David B., and HOVORKA Roman. Closed-loop insulin delivery for treatment of type 1 diabetes. *BMC Medicine*, 2011, 9:120 doi:10.1186/1741-7015-9-120
- [65] BERGMAN R.N., and BUCOLO R.J. Nonlinear metabolic dynamics of the pancreas and liver. *Journal of Dynamic Systems, Measurement, and Control*, 1973, 95, pp. 296-300.
- [66] LI Jiaxu, KUANG Yang, and MASON Clinton C. Modeling the glucose–insulin regulatory system and ultradian insulin secretory oscillations with two explicit time delays. *Journal of Theoretical Biology*, 2006, 242(3), pp. 722–735.
- [67] HOVORKA R., CANONICO V., CHASSIN L.J., HAUETER U., MASSI-BENEDETTI M., ORSINI FEDERICI M., PIEBER T.R., and SCHALLER H.C., SCHAUPP L., VERING T., and WILINSKA M.E. Nonlinear model predictive control of glucose concentration in subjects with type 1 diabetes. *Physiological Measurement*, 2004, 25, pp. 905-920.
- [68] LASSEN Allan Lyngby, NIELSEN Thor Schutt Svane. *Modeling and Simulation of Glucose-Insulin Dynamics*. 94 pages. Bachelor thesis in Electronic, University of Denmark, 2008.
- [69] NAERUM Miriam M.N.. *Model Predictive Control for Insulin Administration in People with Type 1 Diabetes*. 134 pages. Thesis in Electronic, University of Denmark, 2010.
- [70] DALLA MAN Chiara, RIZZA Robert A., and COBELLI Claudio. Meal simulation model of the glucose-insulin system. *IEEE Transactions on Biomedical Engineering*, 2007, 54(10), pp. 1740-1749.
- [71] GARCIA-JARAMILLO Maira Alejandra. *Prediction of Postprandial Blood Glucose under Intra-Patient Variability and Uncertainty and its Use in the Design of Insulin Dosing Strategies for Type 1 Diabetic Patients*. 178 pages. Thesis in Electronic, Universitat de Girona, 2011.

- [72] MAKROGLOU Athena, LI Jiaxu, and KUANG Yang, Mathematical models and software tools for the glucose-insulin regulatory system and diabetes: an overview. *Applied Numerical Mathematics*, 2006, 56(3-4), pp. 559-573.
- [73] PONT Michael J. *Programming Embedded Systems using C*. Addison-Wesley, 2006, 320 pages.
- [74] WEBSTER John G. *Medical Instrumentation Application and Design*. 4th edition, wiley, chapter 10, 2008, 713 pages.
- [75] The accuracy of the GlucoWatch G2 biographer in children with type 1 diabetes: results of the diabetes research in children network (DirecNet) accuracy study, diabetes technology and therapeutics. *Diabetes Research in Children Network*, 2003, 5(5), pp.791-800.
- [76] HOVORKA R. Continuous glucose monitoring and closed-loop systems. *Diabetic Medicine*, 2006, 23(1), pp. 1-12.
- [77] YOO Eun-Hyung, and LEE Soo-Youn, Glucose biosensors: an overview of Use in clinical practice. *Sensors*, 2010, 10(5), pp. 4558-4576.
- [78] CHAMBERS J.P., ARULANANDAM B.P., MATTA L.L., WEIS A., and VALDES J.J. Biosensor recognition elements. *Current Issues in Molecular Biology*, 2008, 10(1-2), pp. 1-12.
- [79] GILBERTSON Roger G., and BUSCH John D. A survey of micro-actuator technologies for future spacecraft missions. *Journal of the British Interplanetary Society*, 1996, 49, pp. 129-138.
- [80] Dr. Fritz Faulhaber GMBH & CO. KG, Germany, Website, www.faulhaber.de, website last accessed May 2014
- [81] Wolfson School of Mechanical and Manufacturing Engineering. *An Introduction to MEMS (Micro-electromechanical Systems)*. Loughborough University, 2002, 56 pages.
- [82] MORETTI Sabrina. In silico experiments in scientific papers on molecular biology. *Science Studies*, 2011, 24(2), pp. 23-42.

- [83] KOVATCHEV Boris P., BRETON Marc, MAN Chiara Dalla, and COBELLI Claudio. In silico preclinical trials: a proof of concept in closed-loop control of type 1 diabetes. *Journal of Diabetes Science and Technology*, 2009, 3(1), pp. 44-55.
- [84] PATEK Stephen D., BEQUETTE B. Wayne, BRETON Marc, BUCKINGHAM Bruce A. In silico preclinical trials: methodology and engineering guide to closed-loop control in type 1 diabetes mellitus. *Journal of Diabetes Science and Technology*, 2009, 3(2), pp. 269-282.
- [85] TONKS David B. A study of the accuracy and precision of clinical chemistry determinations in 170 canadian laboratories. *Clinical Chemistry*, 1963, 9(2), pp. 217-233.
- [86] COX Daniel J., CLARKE William L., GONDER-FREDERICK Linda A., POHL Stephen L., HOOVER Claudia, SNYDER Andrea, ZIMBELMAN Linda, CARTER William R., BOBBITT Sharon, and PENNEBAKER James. Accuracy of perceiving blood glucose in IDDM. *Diabetes Care*, 1985, 8(6), pp. 529-536.
- [87] CLARKE William L., COX Daniel J., GONDER-FREDERICK Linda A., CARTER William R., and POHL Stephen L. Evaluating clinical accuracy of systems of self-monitoring of blood glucose. *Diabetes Care*, 1987, 10(5), pp.622-628.
- [88] DOSHI-VELEZ Finale. The infinite partially observable Markov decision process. *The 24th Annual Conference on Neural Information Processing Systems (NIPS)*, 2009.
- [89] "Draft Guidance for Industry and FDA Staff: Recommendations for Clinical Laboratory Improvement Amendments of 1988 (CLIA) Waiver Applications", U.S. Department of Health and Human Services, September 7, 2005
- [90] Jan S. Krouwer, George S. Cembrowski, "A Review of Standards and Statistics Used to Describe Blood Glucose Monitor Performance", *J Diabetes Sci Technol*. Jan 2010; 4(1): 75–83.
- [91] NASH A., SECKER D., COREY M., DUNN M., and O'CONNOR D.L. Field testing of the 2006 World Health Organization growth charts from birth to 2 years: assessment of hospital undernutrition and overnutrition rates and the usefulness of BMI. *Journal of Parenteral and Enteral Nutrition (JPEN)*, 2008, 32(2), pp. 145-153.

APPENDIX A

This appendix contains the mathematical equations of the minimal model and the dynamic model of the glucose-insulin system.

- Minimal model

The minimal model [11, 12] of the glucose-insulin system is:

$$\frac{dG}{dt} = -[b_1 + X(t)]G(t) + b_1 G_b$$

$$\frac{dX}{dt} = b_2 X(t) + b_3 [I(t) - I_b]$$

$$\frac{dI}{dt} = b_4 [G(t) - b_5] - b_5 [I(t) - I_b]$$

Where

t [min] is time;

$G(0) = b_0$, $X(0) = 0$, $I(0) = b_7 + I_b$, $[G(t) - b_5]^+ = G(t) - b_5$ if $G(t) > b_5$ and 0 otherwise.

$G(t)$ denotes blood glucose concentration at time t [mg/dl]

$I(t)$ insulin blood concentration [pM]

$X(t)$ is an auxiliary function representing insulin-excitable tissue glucose uptake activity

G_b [mg/dl] is the subject's baseline glycemia

I_b is the subject's baseline insulinemia

b_1 – b_6 are various rate constants, and b_0 , b_7 are constants.

- Dynamical model

The model describes glucose concentration changes in blood as depending on spontaneous, insulin-independent net glucose tissue uptake, on insulin-dependent net glucose tissue uptake and on constant baseline liver glucose production.

The dynamic mode [11] of the glucose-insulin system to be studied is:

$$\frac{dG(t)}{dt} = -b_1 G(t) - b_4 I(t) G(t) + b_7, \quad G(t) \equiv G_b \quad \forall t \in [-\infty, 0), \quad G(0) = G_b + b_0$$

$$\frac{dI(t)}{dt} = -b_2 I(t) + b_6 \int_0^\infty \omega(s) G(t-s) ds, \quad I(0) = I_b + b_3 b_0,$$

where

t [min]	is time.
G [mg/dl]	is the glucose plasma concentration.
G_b [mg/dl]	is the basal (pre-injection) plasma glucose concentration.
I [pM]	is the insulin plasma concentration.
I_b [pM]	is the basal (pre-injection) insulin plasma concentration.
b_0 [mg/dl]	is the theoretical increase in plasma concentration over basal glucose concentration at time zero after instantaneous administration and redistribution of the I.V. glucose bolus.
b_1 [min ⁻¹]	is the spontaneous glucose first order disappearance rate constant.
b_2 [min ⁻¹]	is the apparent first-order disappearance rate constant for insulin.
b_3 [pM/(mg/dl)]	is the first-phase insulin concentration increase per (mg/dl) increase in the concentration of glucose at time zero due to the injected bolus.
b_4 [min ⁻¹ pM ⁻¹]	is the constant amount of insulin-dependent glucose disappearance rate constant per pM of plasma insulin concentration.
b_6 [min ⁻¹ pM/(mg/dl)]	is the constant amount of second-phase insulin release rate per (mg/dl) of average plasma glucose concentration per unit time.

b_7 [(mg/dl) min⁻¹] is the constant increase in plasma glucose concentration due to constant baseline liver glucose release.

Noting that Insulin plasma concentration changes are considered to depend on a spontaneous constant-rate decay, due to insulin catabolism, and on pancreatic insulin secretion. The delay term refers to the pancreatic secretion of insulin: effective pancreatic secretion at time t is considered to be proportional to the average value of glucose concentration in the b_5 minutes preceding time t .

APPENDIX B

This appendix contains the implementation of the mathematical model Hovorka in μ Vision Keil program, using a debugger file. The implementation includes also a day life scenarios for a patient.

Hovorka model applied in Keil debugger file

```
Func void hovorka(float we,float uI, float dbas)
{
    wei=we;
    exec("testGB(wei)");
    dD1 = A_G*dbas - D1/tau_G ;
    dD2 = D1/tau_G - D2/tau_G ;
    dS1 = uins - S1/tau_I ;
    dS2 = S1/tau_I - S2/tau_I ;
    dQ1 = -(F_01c + F_R )-x1*Q1+ k_12*Q2 + U_G + EGP_0*(1- x3) ;
    dQ2 = x1*Q1 - (k_12 + x2)*Q2 ;
    dI = U_I/V_I - k_e*I ;
    dx1 = k_b1*I - k_a1*x1 ;
    dx2 = k_b2*I - k_a2*x2 ;
    dx3 = k_b3*I - k_a3*x3 ;
    D1=D1 + h*dD1;
    D2=D2 + h*dD2;
    S1=S1 + h*dS1;
    S2=S2 + h*dS2;
    Q1=Q1 + h*dQ1;
    Q2=Q2 + h*dQ2;
    I=I + h*dI;
    x1=x1 + h*dx1;
    x2=x2 + h*dx2;
    x3=x3 + h*dx3;
}
```

Hovorka model applied in Keil debugger file

```

func void testGB(long bwi)
{
  BW=bwi;
  U_G = D2/tau_G ;
  U_I = S2/tau_I ;
  V_I = 0.12*BW ;
  V_G = 0.16*BW ;
  F_01 =0.0097*BW;
  EGP_0 = 0.0161*BW ;
  uins=0.0954119*BW;
  Gb = Q1/V_G ;
  if(Gb >=4.5){
    F_01c = F_01;}
  else{
    F_01c = F_01*Gb/4.5;}
  if(Gb >=9){
    F_R = 0.003*(Gb - 9)*V_G ;}
  else {
    F_R = 0;
  }
}

```

Day life scenario.

```

signal void Testins()
{
  exec("RELEVENTMOMENTS(8,0,12,0,19,30,30,30)");
  ij=0;
  GB_VAL=0;
  teste=(Q1*18)/12;
  exec("reading3=teste");
  hvalue=0;
  while(1){
    swatch(h); //hsec

```

```

countspan=countspan+h;
ij=countspan;
if (ij!=0){
teste=(Q1*18)/12;
exec("reading3=teste");
if (ij%100==0)
{

GB_VAL=GB_VAL+1;
teste=(Q1*18)/12 ;
exec("reading3=teste");
exec("S1=S1+dose*1000");
exec("dose=0");
printf("%f ",teste);
}
exec("reading3=teste");
printf("%f ",teste);
if (ij%100 ==0)
{
//printf("%f\n",S1);

printf("\n");
}
}

if (t1<countspan && t2>=countspan) //Midnight to first insulinshot
{
exec("hovorka(75,uins,0)");
}

if (t2<countspan && t3>=countspan) //Insulinshot before breakfast
{
if (bbit==0){
exec("initial(binsulin)");
bbit=1;
}
exec("hovorka(75,uins,0)");
}

```

```
}

if(t3<countspan && t4>=countspan)                // Breakfast start
{
dch=bCHO/(180*eatingTime)*1000;
exec("hovorka(75,uins,dch)");
}

if(t4<countspan && t5>=countspan)                // Breakfast stop
{
exec("hovorka(75,uins,0)");
}

if (t5<countspan && t6>=countspan)                // Insulinshot before lunch
{
if (lbit==0){
exec("initial(linsulin)");
lbit=1;
}
exec("hovorka(75,uins,0)");
}

if (t6<countspan && t7>=countspan)                // lunch start
{
dch=lCHO/(180 * eatingTime )*1000;
exec("hovorka(75,uins,dch)");
}

if (t7<countspan && t8>=countspan)                //lunch stop
{
exec("hovorka(75,uins,0)");
}

if (t8<countspan && t9>=countspan)                // Insulinshot before dinner
{
```

```
if (dbit==0){
exec("initial(dinsulin)");
dbit=1;
}
exec("hovorka(75,uins,0)");
}

if (t9<countspan && t10>=countspan)                // dinner start
{
dch=dCHO/(180 * eatingTime )*1000;
exec("hovorka(75,uins,dch)");
}

if (t10<countspan && t11>countspan)                // dinner stop
{
exec("hovorka(75,uins,0)");
}

if (t11<=countspan)
{
countspan=0;
exec("EXIT");
}
} //end while
} //end function
```

APPENDIX C

This appendix contains the implementation of the percentile function in VB.Net program, the median function, the Standard Deviation function and the error grid implementation in Matlab program.

Percentile function

```
Public Function percentile(ByVal sortedData() As Decimal, ByVal p As Double) As Double

    If p >= 100.0 Then

        Return sortedData(sortedData.Length - 1)

    End If

    Dim position As Double = CDb1(sortedData.Length + 1) * p / 100.0

    Dim leftNumber As Double = 0.0, rightNumber As Double = 0.0

    Dim n As Double = p / 100.0 * (sortedData.Length - 1) + 1.0

    If position >= 1 Then

        leftNumber=sortedData(CInt(Math.Truncate(System.Math.Floor(n))) - 1)

        rightNumber = sortedData(CInt(Math.Truncate(System.Math.Floor(n))))

    Else

        leftNumber = sortedData(0)

        rightNumber = sortedData(1)

    End If

    If leftNumber = rightNumber Then

        Return leftNumber

    Else

        Dim part As Double = n - System.Math.Floor(n)

        Return leftNumber + part * (rightNumber - leftNumber)

    End If

End Function
```

Median function

```

Public Function median1(ByVal NumericArray() As Integer) As Double

    Dim numberCount As Integer = NumericArray.Count

    Dim halfIndex As Integer = NumericArray.Count \ 2

    Dim sortedNumbers = NumericArray.OrderBy(Function(n) n)

    Dim median As Double

    If (numberCount Mod 2 = 0) Then

        Median = (sortedNumbers.ElementAt(halfIndex) +
sortedNumbers.ElementAt(halfIndex - 1)) / 2

    Else

        median = sortedNumbers.ElementAt(halfIndex)

    End If

    Return median

End Function

```

Standard deviation function

```

Public Function getStandardDeviation(ByVal doubleList() As
Integer) As Double

    Dim average As Decimal = doubleList.Average()

    Dim sumOfDerivation As Double = 0

    For Each value As Double In doubleList

        sumOfDerivation += (value) * (value)

    Next

    Dim sumOfDerivationAverage As Double=sumOfDerivation / doubleList.Count

    Return Math.Sqrt(sumOfDerivationAverage - (average * average))

End Function

```

Error grid implementation

```

function [] = clarke4()
% ----- Print figure flag -----
PRINT_FIGURE = true;
% ----- Plot Clarke's Error Grid -----
h = figure;
strpat7c=[90.374962 90.381195 90.387421 90.393639 .....]
n=length(strpat7c);

```



```

for i=1 : 1: n
    str1(i)=strpat7c(i);
end
prct25p1 = prctile(str1,2.5);
prct975p1 = prctile(str1,97.5);
if (prct25p1<50)
    prct25p1=50;
end
if (prct975p1<110)
    prct975p1=110;
end

xlabel('Minimum BG (2.5th percentile)');
ylabel ('Maximum BG (97.5th percentile)');
title('Variability-grid analysis of patients');
set(gca,'XLim',[50 110]);
set(gca,'YLim',[110 400]);
set(gca,'XDir','reverse')
axis square
hold on

%Upper C
fill([90 110 110 90],[300 300 400 400],'y');
text(105,350,'Upper C','FontSize',12);

%Upper Bzone
plot([90 90],[185 300],'k-')
plot([90 180],[300 300],'k-')
fill([90 110 110 90],[180 180 300 300],[7/255 135/255 0/255]);
text(105,250,'Upper','FontSize',12);
text(105,230,'B-zone','FontSize',12);

%A-zone
fill([90 110 110 90],[110 110 180 180],'g');
text(105,150,'A-zone','FontSize',12);

%Upper D
fill([90 70 70 90],[300 300 400 400],[1 0.5 0.2]);
text(85,350,'Upper D','FontSize',12);

%Bzone
fill([70 90 90 70],[180 180 300 300],[7/255 135/255 0/255]);
text(85,250,'B-zone','FontSize',12);

%Lower B-zone
% plot([70 70],[110 180],'k-')           % Horizontal regression line
% plot([90 70],[180 180],'k-')
fill([70 90 90 70],[110 110 180 180],[7/255 135/255 0/255]);
text(85,150,'Lower','FontSize',12);
text(85,130,'B-zone','FontSize',12);

%E-zone

```

```
fill([50 70 70 50],[300 300 400 400],'r');
text(65,350,'E-zone','FontSize',12);

%Lower D
fill([50 70 70 50],[180 180 300 300],[1 0.5 0.2]);
text(65,250,'Lower D','FontSize',12);

%Lower C
fill([50 70 70 50],[110 110 180 180],'y');
text(65,150,'Lower C','FontSize',12);

set(h, 'color', 'white');           % sets the color to white
% Specify window units
set(h, 'units', 'inches');

plot(prct25p1,prct975p1,'ko','MarkerSize',4,'MarkerFaceColor','k','MarkerEdgeColor','k');
```